

DAS VIRTUELLE BAUWERK ALS
INFORMATIONSUMGEBUNG FÜR DIE
PLANUNG IM BESTAND
ZUR ORGANISATION UND STRUKTURIERUNG EINER
DIGITALEN BAUWERKSDOKUMENTATION

DISSERTATION

zur Erlangung des akademischen Grades
DOKTOR-INGENIEUR

an der Fakultät Bauingenieurwesen
der
Bauhaus-Universität Weimar
vorgelegt von

Dipl.-Ing. Katrin Wender
geboren am 17. Juli 1977 in Mühlhausen

Gutachter:

Prof. Dr.-Ing. habil. Reinhard Hübler, Bauhaus-Universität Weimar
Prof. Dr.-Ing. Frank Petzold, Bauhaus-Universität Weimar
Prof. Dr.-Ing. Petra von Both, Technische Universität Karlsruhe

Eingereicht am: 26. November 2008

Tag der Disputation: 28. Mai 2009

VORWORT

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin an der Professur Informations- und Wissensverarbeitung an der Bauhaus-Universität Weimar. Das bearbeitete Thema war dabei Gegenstand des von der Deutschen Forschungsgemeinschaft (DFG) geförderten Einzelprojektes „Modulare, dynamische Recherchewerkzeuge zur Unterstützung der kooperativ-integrativen Entscheidungsfindung im Bauwerkslebenszyklus“.

Ich möchte an dieser Stelle all denjenigen ganz herzlich danken, die – auf unterschiedliche Weise – zum Gelingen dieser Arbeit beigetragen haben.

In erster Linie danke in meinem Mentor Prof. Reinhard Hübler für die wissenschaftliche Betreuung sowie für seine permanente Diskussionsbereitschaft und konstruktive Kritik, mit der er das Entstehen der Arbeit von den ersten Gedanken an begleitete. Er hat mir darüber hinaus die nötigen Freiräume für das Schreiben dieser Arbeit geschaffen und stand mir während dieser Zeit immer wieder anspornend und mit fachlichem Rat zur Seite.

Mein Dank gilt weiterhin Prof. Petra von Both und Prof. Frank Petzold für die Begutachtung der Arbeit und die Diskussion der Problematik aus unterschiedlichen fachlichen Perspektiven.

Den Kollegen an den Lehrstühlen Informations- und Wissensverarbeitung, Informatik in der Architektur und Informatik im Bauwesen danke ich für die freundschaftliche Arbeitsatmosphäre und den offenen fachlichen Austausch. Für die Unterstützung beim Korrekturlesen sowie wertvolle fach-

liche und redaktionelle Hinweise danke ich insbesondere Anke Schwanck, Christina Wickler, Christian Koch und Daniel G. Beer. Frau Dr. Heidemarie Schirmer danke ich für ihr Engagement und die gute Zusammenarbeit bei der Veröffentlichung dieser Arbeit.

Schließlich danke ich meiner Familie und meinen Freunden, die meine temporäre geistige und körperliche Abwesenheit mit gelassener Ruhe respektiert haben, und ohne deren persönliche Unterstützung diese Arbeit wohl kaum möglich gewesen wäre. Ein besonderer Dank gilt Lars Reimer für seine Ruhe beim Zuhören zu jeder erdenklichen Tageszeit, vor allem aber für seine ständige Ermunterung, gemeinsam nach vorn und auf das Ganze zu schauen.

Weimar, im Juli 2009

Katrin Wender

*"Perfection is achieved, not when there is nothing more to add,
but when there is nothing left to take away."¹*

¹ Antoine de Saint-Exupéry (1900-1944), französischer Flieger und Schriftsteller

KURZFASSUNG

Objektorientierte Bauwerksmodelle sind derzeit Gegenstand umfangreicher Forschungsaktivitäten zur rechnerinternen Verwaltung bauwerksbezogener Informationen. Ein in diesem Rahmen diskutierter Ansatz ist die Realisierung eines virtuellen rechnerinternen Bauwerks in Form eines variablen Verbundes fachspezifischer objektorientierter Modelle. Diese Organisationsform der rechnerinternen Repräsentation eignet sich einerseits aufgrund ihrer Flexibilität sehr gut für eine lebensphasenübergreifende Fortschreibung als digitale Bauwerksakte. Eine solche Bauwerksakte bildet eine wichtige Informationsgrundlage für die Planung von Instandhaltungs-, Modernisierungs-, Umbau- oder Erweiterungsmaßnahmen in späteren Lebensphasen des Bauwerks. Andererseits erschwert die dezentrale Organisationsform des Modellverbundes jedoch die Informationssuche, was in erster Linie durch das Vorhandensein multipler Repräsentationen einzelner Realweltobjekte sowie die Komplexität der am Verbund beteiligten Modellschemata bedingt ist.

Gegenstand der vorliegenden Arbeit ist die Entwicklung eines generischen Systemkerns als Basis für eine projektbezogen konfigurierbare Informationsumgebung, die insbesondere in frühen Projektphasen einen handhabbaren Zugang zu den im Modellverbund verwalteten Informationen bereitstellen kann. Der vorgeschlagene Lösungsansatz erweitert den Modellverbund um eine Erschließungsstruktur, in der die einzelnen Elemente der individuellen baulich-räumlichen Struktur durch eindeutige Identifikatoren vertreten werden. Mit den Identifikatoren werden jeweils die fachspezifischen Repräsentationen des Objektes verknüpft und sind somit von einem zentralen

Einstiegspunkt aus erreichbar. Der generische Systemkern definiert eine objektorientierte Datenstruktur zur Verwaltung der jeweils projektbezogen auszurägenden Erschließungsstruktur.

Die Interaktion mit dem erschlossenen Informationsraum erfordert eine entsprechende Nutzerschnittstelle, die in der Lage ist, nicht vorhersehbaren, spontan entstehenden Informationsbedarf zu bedienen. Darüber hinaus soll sie konfigurierbar sein bezüglich der unterstützten Suchstrategien. Der Lösungsansatz sieht eine hierarchische Organisationstruktur der Nutzerschnittstelle vor, die eine modulare Erweiterung ermöglicht. Ein entsprechender Kern der Nutzerschnittstelle wird als objektorientiertes Framework spezifiziert.

Die Erschließungsstruktur und die Nutzerschnittstelle werden unter Anwendung des objektorientierten Paradigmas entwickelt und mit Hilfe einer formalen Notation auf implementierungsunabhängiger Ebene beschrieben. Anhand exemplarischer Umsetzungen kritischer Systemteile wird die prinzipielle Realisierbarkeit des beschriebenen Systems nachgewiesen.

ABSTRACT

Object oriented building models are subject of comprehensive research activities aiming at the computer-supported management of building related information. An interesting approach discussed in this field is to set up a virtual digital building as a variable compound of several discipline specific object oriented models. Because of its flexible form of organization such a digital representation can be easily modified and enhanced throughout the entire life cycle of a building, and therefore it is well suited as a basis for a digital cross-life cycle documentation of a building.

Such a documentation can serve as an important information base for the planing of any reconstruction work on the particular building. However the decentralized organization of the model compound does not support an intuitive navigation and information seeking. The model schemata can be very complex and a particular real world object may have multiple digital representations spread among the discipline specific models.

The work presents the development of a generic core of an adaptable information seeking environment which focuses especially at supporting cross-disciplinary information needs in early project phases. The approach proposed in this work enhances the model compound concept by two additional layers: the navigation structure and the user interface. The navigation structure overlays the model compounds and serves as its table of contents. Each real world object is represented in the navigation structure by an unique identifier. The multiple discipline specific repre-

sentations of a particular real world object are linked to its identifier and therefor accessible from a central entry point. Within the generic core an object oriented data structure is defined, that allows for the storage and management of project specific instances of the navigation structure.

For information seeking planners will have to interact with the information space containing the model compound and the navigation structure. Therefore the user interface has to support any kind of unpredictable and spontaneous information need. Also it has to be adaptable in terms of the searching strategies it supports. The proposed system concept bases on an hierarchical organization which allows for configuration by changing and/or adding of modules. The generic core defines an object oriented framework for the individual implementation of the user interface.

The generic core is formally specified on an implementation-independent level. Several experimental prototypes prove the technical feasibility of critical aspects of the system concept.

INHALTSVERZEICHNIS

Vorwort	iii
Kurzfassung	vii
Abstract	ix
Abkürzungsverzeichnis	xxv
1. Einführung	1
1.1. Problemstellung	2
1.2. Zielsetzung	6
1.3. Ausgangspunkte und Vorgehensweise	7
1.4. Aufbau der Arbeit	13
2. Planung im Kontext bestehender Bausubstanz	19
2.1. Begriffe	20
2.2. Nachhaltiges Planen	21
2.3. Bereitstellung planungsrelevanter Informationen	25
2.4. Spezialisierung und Rollenverteilung im Planungsprozess	28
2.5. Computergestützte Werkzeuge für die Planung im Bestand	30
2.6. Randbedingungen und Konsequenzen	33
3. Stand der Bauwerksmodellierung	35
3.1. Abgrenzung	35
3.2. Begriffe	36

3.2.1. Modelle	36
3.2.2. Die Unified Modelling Language - UML	39
3.3. Produktmodelle im Bauwesen - Bauwerksmodelle	40
3.3.1. Modellierungsparadigma	41
3.3.2. Modellschemata und Modelldaten	42
3.4. Der Produktmodellansatz als Methode der Integration	44
3.5. Merkmale und Ansätze integrierter Informationssysteme	46
3.5.1. Art der Kopplung	47
3.5.2. Entwurfsstrategie	48
3.5.3. Ort der Datenspeicherung	49
3.6. Integrationsansätze im Bauwesen	49
3.7. Dynamische Bauwerksmodelle	59
3.8. Konsequenzen für eine digitale Bauwerksdokumentation	66
4. Entwerfen als kreatives Problemlösen	69
4.1. Abgrenzung	69
4.2. Begriffe	72
4.3. Zum Vorgang des kreativen Problemlösens	74
4.3.1. Phasen des Iterationsprozesses	74
4.3.2. Varianten des Iterationsprozesses entsprechend ver- schiedener Barrieretypen	79
4.4. Einflussfaktoren beim Problemlösen	87
4.4.1. Flexibilität des Denkens	88
4.4.2. Formale und informale Lösungsstrategien	90
4.4.3. Wiederverwendung von Entwurfswissen	91
4.5. Entwicklung mentaler Modelle als Hauptaufgabe des Entwer- fens	93
4.6. Konsequenzen für entwurfsunterstützende Werkzeuge	96
5. Informationssuche	99

5.1. Abgrenzung	99
5.2. Begriffe	101
5.2.1. Informationsraum und Navigation	101
5.2.2. Direkte Manipulation	103
5.2.3. Informationsumgebung	103
5.2.4. Informationsproblem	105
5.3. Informationssuche als Problemlöseprozess	106
5.3.1. Teilprozesse der Informationssuche	106
5.3.2. Vorgehensweisen bei der Suche (Suchstrategien) . . .	109
5.3.3. Wechselwirkungen zwischen Problemklassen und Such- strategien	113
5.4. Organisation mentaler Repräsentationen	115
5.4.1. Organisationsprinzipien	115
5.4.2. Orientierung im Entwurfs- und Planungsgegenstand Bauwerk	117
5.5. Konsequenzen für die entwurfsbezogene Informationssuche .	119
6. Problemdefinition und Gesamtkonzept	121
6.1. Begriffe	122
6.2. Anforderungen und Annahmen	122
6.3. Defizite der bestehenden Situation	126
6.3.1. Projektbezogene Organisation des Modellverbundes .	126
6.3.2. Generischer Modellzugriff	129
6.3.3. Komplexität von Bauwerksmodellschemata	131
6.3.4. Präsentationsformen und Interaktionsmechanismen .	134
6.4. Verwendete Analogien	135
6.4.1. Informationsumgebungen im Dokumentretrieval . . .	136
6.4.2. Offene Hypermedia-Systeme	140
6.5. Das virtuelle Bauwerk als Informationsumgebung	145
6.6. Abgrenzung des betrachteten Sachverhaltes	150

7. Erschließung des Informationsraums	155
7.1. Informationscontainer und Eigenschaftslinks	156
7.1.1. Informationscontainer als Bausteine der objektbezo- gen Ordnungsstruktur	156
7.1.2. Unterstützung der Navigation innerhalb komplexer Modellschemata durch Eigenschaftslinks	162
7.2. Konzeptuelles Schema der Erschließungsstruktur	170
7.2.1. Relevante Konzepte	170
7.2.2. Klassenhierarchie	175
7.2.3. Strukturelle Beziehungen zwischen Informationscon- tainern	176
7.2.4. Zuordnung domänenspezifischer Repräsentationen . .	178
7.2.5. Zuordnung von Szeneobjekten und Klassifikationssys- temen	180
7.2.6. Eigenschaftslinks	180
7.3. Formale Repräsentation	183
7.3.1. Ontologien und Semantic Web Technologien	184
7.3.2. Beschreibung der Erschließungsstruktur mit OWL . .	185
7.4. Diskussion	188
8. Koordination von Recherchemodulen	191
8.1. Exemplarisches Szenario einer Recheresitzung	192
8.2. Grundlagen objektorientierter Softwareentwicklung	200
8.2.1. Methoden der Softwareentwicklung	201
8.2.2. Wiederverwendung von Entwurfswissen	202
8.3. Systemkonzept des Rechercheframeworks	207
8.3.1. Vorüberlegungen	208
8.3.2. Strukturierung der Rechereschnittstelle	210
8.3.3. Erzeugen von Viewerexemplaren und Integration neuer Viewerklassen	220

8.3.4. Zentraler Zugriff auf die Erschließungsstruktur und den Modellverbund	232
8.3.5. Alphanumerische Auswertung von Referenzknoten . .	235
8.4. Implementierung des Rechercheframeworks	239
8.5. Diskussion	244
9. Fazit	247
9.1. Wertung	249
9.2. Ausblick	251
A. Übersicht verwendeter Muster	271
B. Methoden der Klasse NaviStructure zum Zugriff auf die Er- schließungsstruktur	287
C. Übersicht zu Prozessorentypen und gültige Nachrichten	291
D. Prototypen	293
D.1. Erschließungsstruktur	293
D.2. Property Link Manager	297
D.3. Navi Model Editor	299
D.4. ISeE – Information Seeking Environment, Framework	302
E. Ehrenwörtliche Erklärung	305
F. Über die Autorin	307

INHALTSVERZEICHNIS

ABBILDUNGSVERZEICHNIS

1.1. Modellzugriff über den Browser der Modellierungsplattform	3
1.2. Typische Arbeitsumgebung eines Architekten	5
1.3. Aufbau der Arbeit	14
2.1. Investitionen am Immobilienmarkt 2003, nach [Hommerich u. a., 2005, S. 31]	20
2.2. Relevanz von Entwurfsentscheidungen früher Planungspha- sen nach [Steinmann, 1997, S. 5]	23
2.3. Vorgehensweise bei Modernisierungsprojekten nach [Wiech- mann, 1981, S. 97]	24
2.4. Brüche im Informationsfluss nach [Gessmann, 2005, S. 2]	28
2.5. Klassifikation entwurfsunterstützender Werkzeuge nach [Stein- mann, 1997, S. 16]	31
3.1. Integrationsrichtungen nach [Bakkeren u. Tolman, 1995]	46
3.2. Prinzip von Peer-DBMS	57
3.3. Klassendiagramm der Ebenen dynamischer Modelle	61
3.4. Verbund einzelner Modellverwaltungssysteme	61
3.5. Konflikt bei dynamischer Modellerweiterung	64
3.6. Modellverbund mit zentralem IFC-Peer	67
4.1. Entwurfsspirale nach Asimov, erweitert um die Externalisie- rung mentaler Modelle	77
4.2. Iterationsschleifen im Entwurfsprozess	77

4.3. Klassifikation von Barrieretypen nach [Dörner, 1976]	80
4.4. Aktivitätsdiagramm: Ablauf beim Lösen von Interpolationsproblemen	82
4.5. Aktivitätsdiagramm: Ablauf beim Lösen von Syntheseproblemen mit Hilfe von Analogien	84
4.6. Aktivitätsdiagramm: Ablauf beim Lösen dialektischer Probleme	86
4.7. Wechselwirkung mentaler und externalisierter Repräsentationen	94
5.1. Informationsumgebung, Informationsraum und Schnittstelle	105
5.2. Übergänge zwischen den Teilprozessen der Informationssuche nach [Marchionini, 1997, S. 50]	107
5.3. Explorative Suche zur Klärung von Informationsproblemen nach [Taube, 1998, S. 158] auf Basis der Problemklassifikationsmatrix nach [Dörner, 1976]	114
6.1. Multiple Repräsentationen ohne expliziten Zusammenhang .	127
6.2. Darstellung eines IfcWindow-Exemplars im generischen MVS-Browser	130
6.3. Zuordnung von Materialien zu einem Wandobjekt in IFC (UML-Klassen-/Objektdiagramm)	133
6.4. Ebenen einer Informationsumgebung zum Dokumentretrieval	137
6.5. Begriffsbildung und Zuordnung von Bezeichnungen in einer Dokumentationssprache	139
6.6. Referenzarchitektur für OHS nach [Anderson u. a., 1999] . .	142
6.7. UML-Klassendiagramm der FOHM-Konzepte	144
6.8. Vorgeschlagene Systemstruktur in Analogie zum Aufbau einer Retrievalumgebung	146

6.9. Bezeichnungen für die Komponenten der Informationsumgebung (fett) und deren Aufgaben (kursiv)	146
6.10. Objektbezogene Ordnungsstruktur und Eigenschaftslinks	147
6.11. Abgrenzung der Arbeitsschwerpunkte	153
7.1. Bereitstellung von Einstiegspunkten durch Knoten und Verknüpfungen	157
7.2. Konkrete Ausprägung eines Informationscontainers für eine Wand (informale Darstellung)	159
7.3. Knoten-Kanten-Notation für ein Informationscontainer-Exemplar	160
7.4. Repräsentation einer Wand als Menge verknüpfter Objekte (UML-Objektdiagramm)	163
7.5. IFC-Objektstruktur zur Verknüpfung eines Wandobjektes mit korrespondierenden Tragwerkselementen (UML-Klassendiagramm)	165
7.6. Knoten-Kanten-Notation für ein Exemplar eines Eigenschaftslinks	167
7.7. Knoten- und Kantentypen zur Abbildung der Erschließungsstruktur	171
7.8. Klassen des ES-Schemas (UML-Klassendiagramm)	172
7.9. Abbildung von Beziehungen zwischen Informationscontainern (UML-Klassendiagramm)	176
7.10. Zuordnung domänenspezifischer Repräsentationen (UML-Klassendiagramm)	178
7.11. Zuordnung von Szeneobjekten (UML-Klassendiagramm)	181
7.12. Zuordnung von Typen aus externen Klassifikationen (UML-Klassendiagramm)	181
7.13. Eigenschaftslinks (UML-Klassendiagramm)	181

7.14. Temporäre Ausprägung von Eigenschaftslinks für Modelldaten (UML-Klassendiagramm)	182
7.15. Organisation der OWL-Dokumente zur Repräsentation der Erschließungsstruktur	186
8.1. Exemplarisch betrachtetes Bürogebäude	193
8.2. Verfügbarer Informationsraum des Szenarios	193
8.3. Übersicht der Dokumentation für das <i>Bürogebäude Kastanienallee</i>	195
8.4. 3D-Szene nach dem Start des Viewermoduls	196
8.5. 3D-Szene mit Kennzeichnung des Tragsystems	197
8.6. 3D-Szene mit Kennzeichnung der Räume mit baulichen Schäden	198
8.7. Detailsicht für ein Informationscontainer-Exemplar (schematisch)	199
8.8. Detailsicht für ein Datenobjekt eines Domänenmodells (schematisch)	200
8.9. Informale Darstellung der Viewer-Manager-Struktur	214
8.10. Klassendiagramm der Viewer-Manager-Struktur	215
8.11. Sequenzdiagramm: Starten des ersten Viewer -Agenten	216
8.12. Sequenzdiagramm: Starten eines Viewers und eines neuen Managers für eine gewählte Teilmenge	217
8.13. Sequenzdiagramm: Erzeugen und Darstellen einer Menge farbig zu kennzeichnender Informationscontainer	218
8.14. Sequenzdiagramm: Starten einer Detail-Darstellung zu einem einzelnen Informationscontainer	219
8.15. Informale Darstellung der Fabrik-Manager-Struktur	222
8.16. Klassendiagramm: Manager- und Fabrikagenten	223
8.17. Sequenzdiagramm: Anmelden eines neuen Fabrikagenten und Erzeugen eines neuen Vieweragenten	225

8.18. Informale Darstellung der Gesamtstruktur des Frameworks	227
8.19. Klassendiagramm zur Gesamtstruktur	228
8.20. Klassendiagramm: Fabriken und erzeugte Agententypen	229
8.21. Sequenzdiagramm: Starten einer Recheresitzung	230
8.22. Sequenzdiagramm: Anfordern eines neuen Vieweragenten	231
8.23. Informale Darstellung zum zentralen Zugriff auf den Informationsraum	232
8.24. Klassendiagramm: Zugriff auf die Erschließungsstruktur über die Klassen <code>NaviStructure</code> und <code>NaviStructureManager</code>	234
8.25. Sequenzdiagramm: Aufbau der Übersicht	235
8.26. Sequenzdiagramm: Ermitteln darzustellender Inhalte	236
8.27. Klassendiagramm: Zuordnung von Prozessoren zu Referenzknotenklassen	237
8.28. Klassendiagramm: Zuordnung von Viewerdiensten zu Referenzknotenklassen (exemplarisch)	238
8.29. Agentenhierarchie nach dem Start der Informationsumgebung	240
8.30. Session Manager und Jade Remote Agent Management GUI	241
8.31. Agentenhierarchie nach dem Start des 3D-Viewers	242
8.32. Benutzeroberflächen einzelner aktiver Agenten	243
A.1. Komponenten des <i>Model-View-Controller</i> -Musters (UML-Klassendiagramm)	273
A.2. Interaktion der <i>MVC</i> -Komponenten (UML-Sequenzdiagramm)	274
A.3. Komponenten des <i>Presentation-Abstraction-Control</i> -Musters (UML-Klassendiagramm)	277
A.4. Interaktion der <i>PAC</i> -Komponenten (UML-Sequenzdiagramm)	278
A.5. Komponenten des Musters <i>Fabrikmethode</i> (UML-Klassendiagramm)	280
A.6. Komponenten des Musters <i>Prototyp</i> (UML-Klassendiagramm)	282

D.1. Bearbeitung der Erschließungsstruktur in Protégé (gelb: Schema, violett: Daten)	293
D.2. Benutzeroberfläche: Übersicht zu definierten Eigenschaftslinks	297
D.3. Kontrollausgabe einer erzeugten Anfrage	298
D.4. Übersichts- und Arbeitsfenster des Editors	299
D.5. 3D-Fenster zur Auswahl von Szeneobjekten	300
D.6. Fenster zur Auswahl domänenspezifischer Repräsentationen (IFC-Objekte)	300
D.7. Grafische Oberfläche des Jade Remote Manager Agenten . .	303
D.8. Grafische Oberfläche des <code>SessionManager</code> -Agenten	303

TABELLENVERZEICHNIS

2.1. Kosten für die Bauaufnahme nach [Schmitz u. Krings, 2004, S. 35]	26
3.1. Bezeichnungen der unterschiedlichen Modellierungsebenen	43
3.2. Verwendete Bezeichnungen für die Modellierungsebenen	44
3.3. Bezeichnungen der Ebenen in verschiedenen Modellierungsansätzen	51
3.4. Technische Merkmale der Integrationsansätze	54
4.1. Klassen von Entwurfsproblemen	70
4.2. Phasen des Problemlösevorgangs bei verschiedenen Autoren	75
5.1. Teilprozesse der Informationssuche nach [Marchionini, 1997]	106
5.2. Merkmale analytischer und explorativer Suchstrategien	111
5.3. Analytische und explorative Suchstrategien - resultierende Anforderungen	112
7.1. Mögliche Darstellungsprinzipien für die Knoten- und Assoziationsklassen	173
7.2. Beziehungsklassen und deren Bedeutung	178
7.3. Zulässige Werte des <code>type</code> -Attributs zur Kennzeichnung der logischen Zuordnung zwischen Informationscontainern und domänenspezifischen Repräsentationen	180
A.1. Schablone zur Musterbeschreibung	272

TABELLENVERZEICHNIS

ABKÜRZUNGSVERZEICHNIS

ACL	Agent Communication Language
AEC	Architecture/Engineering/Construction
AKO	Arbeitskreis Objekte
AP	Ausgangspunkt
AVA	Ausschreibung, Vergabe, Abrechnung
BGF	Brutto-Grundfläche
BRI	Brutto-Rauminhalt
CAAD	Computer Aided Architectural Design
CAD	Computer Aided Design
CAFM	Computer Aided Facility Management
CHI	Content Handler Interface
CIM	Computer Integrated Manufacturing
CSF	Client-Side Function
DBMS	Datenbankmanagementsystem
ES	Erschließungsstruktur
EUR	Euro (Währungseinheit)
FIPA	Foundation for Intelligent Physical Agents
FM	Facility Management
FOHM	Fundamental Open Hypermedia Model
GUI	Graphical User Interface
HOAI	Honorarordnung für Architekten und Ingenieure
HTML	Hypertext Markup Language
HVAC	Heating Ventilation Air Conditioning

IAI	International Alliance for Interoperability, deutsch: Industrieallianz für Interoperabilität
IFC	Industry Foundation Classes
IFD	International Framework for Dictionaries
JADE	Java Agent Development Framework
KI	Künstliche Intelligenz
LP	Leistungsphase (der HOAI)
MVS	Modellverwaltungssystem
OHP	Open Hypermedia Protocol
OHS	Offenes Hypermedia-System
OMG	Object Management Group
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFS	RDF Schema
TGA	Technische Gebäudeausstattung
UML	Unified Modelling Language
W3C	Word Wide Web Consortium
WE	Wohneinheiten
WWW	Word Wide Web
XML	Extensible Markup Language

1. EINFÜHRUNG

Bauwerke, insbesondere Gebäude, sind im Verlauf ihres Lebenszyklus wiederholt baulichen Änderungen unterworfen, die eine Erhaltung der bestehenden Bausubstanz und/oder deren Anpassung an veränderte Nutzungsanforderungen zum Ziel haben. Die Planung derartiger Maßnahmen wird allgemein als Planung im Kontext bestehender Bausubstanz bzw. *Planung im Bestand* bezeichnet. Dazu vgl. [SFB 524, 2005, 2006; GISMO Projekt, 2008].

*Erhaltung und
Anpassung
bestehender
Bausubstanz*

Die frühen Planungsphasen solcher Projekte, in denen das Gesamtkonzept für das Projekt entwickelt wird, haben maßgeblichen Einfluss auf die Nachhaltigkeit der veranlassten Maßnahmen in ihrer Gesamtheit [Pfeiffer u. a., 2001; Steinmann, 1997]. Gleichzeitig bedeutet Planen im Bestand häufig die Planung von Eingriffen in ein komplexes Gesamtsystem aus bestehendem Bauwerk (Gebäude) und bestehenden Nutzungsabläufen [Vester, 1999], wobei eine vollständige Erfassung dieses Gesamtsystems meist nicht möglich ist. Informationen zur bestehenden Bausubstanz werden planungsbegleitend benötigt und, falls nicht verfügbar, erfasst. In der Regel werden parallel zur Konkretisierung bzw. Präzisierung des Planungskonzeptes schrittweise detailliertere Informationen zum baulichen Bestand benötigt. Dazu vgl. u. a. [Wiechmann, 1981; Petzold, 2001; Thurow, 2004].

*Entwicklung des
Planungskonzeptes
für
„Bestandsprojekte“*

Die vorliegende Arbeit widmet sich konkret der Unterstützung des Entwurfs, d.h. der frühen Phasen der Planung im Bestand, in denen das Gesamtkonzept für ein Projekt gefunden und formuliert werden muss.

1.1. PROBLEMSTELLUNG

*Datenverwaltung in
variablen
Modellverbund* In den vergangenen Jahren entwickelte Ansätze zur Datenverwaltung und Datenintegration in Bauprojekten schlagen einen variablen Verbund einzelner fachspezifischer, objektorientierter Modelle vor. Die Struktur des Verbundes sieht das Austausch oder Hinzufügen von Modellen entsprechend projektspezifischer Anforderungen vor und ist damit flexibel an unterschiedliche Projektsituationen und Bauwerkstypen anpassbar. Darüber hinaus ist ein Fortschreiben und Ergänzen über den gesamten Lebenszyklus eines Bauwerks möglich. Dazu vgl. u. a. [Hübler u. a., 2003; Willenbacher, 2002; Hauschild, 2003; Rüppel u. a., 2006, 2007].

„Virtuelles Bauwerk“ Der Modellverbund realisiert einen logischen Gesamtdatenbestand in Form eines „virtuellen Bauwerks“, der während eines Bauprojektes als Integrationsplattform für verteilte Bauplanungsprozesse dienen kann [Hübler, 2005]. Auf der Basis eines solchen Modellverbundes ist zugleich auch die Realisierung der Datenverwaltung für eine lebenszyklusübergreifende Bauwerksdokumentation oder digitale Bauakte, wie sie u. a. in [Gessmann, 2005; GISMO Projekt, 2008] gefordert wird, möglich.

*Multiple
Repräsentationen auf
Datenebene* Die einzelnen am Verbund beteiligten Modelle stellen jeweils fachspezifische Sichten auf den gemeinsamen Planungsgegenstand, das Bauwerk, dar. Dies hat zur Folge, dass zu den einzelnen Elementen eines Bauwerks, wie Räumen oder Bauteilen, multiple fachspezifische Repräsentationen verteilt über den Modellverbund existieren, wodurch das Auffinden aller für ein Element (z. B. Stütze-012 oder Raum-007) verfügbaren Informationen erschwert wird. In diesem Sinne kann von einem solchen Modellverbund nur bedingt als von einem *virtuellen Bauwerk* gesprochen werden.

*Zugriff
auf/Darstellung von
Modelldaten* Für den Zugriff auf die in einem variablen Modellverbund verwalteten Informationen ergeben sich grundsätzlich zwei Möglichkeiten. Die erste besteht in der Verwendung der Fachapplikationen, die zur Erstellung und

Bearbeitung der einzelnen Modelle dienen. Fachapplikationen sind in der Lage, die spezifische Semantik „ihrer“ Modelle zu interpretieren und für den Nutzer entsprechende Darstellungen zu generieren. Jedoch ist auf diese Weise ein modellübergreifender Zugriff auf Informationen nicht möglich. Die Verwendung einer generischen Nutzerschnittstelle bietet eine zweite Möglichkeit für den Modellzugriff. Diese Nutzerschnittstelle kann bzw. soll die spezifische Semantik der Modelle aufgrund der Variabilität des Verbundes nicht kennen. Sie arbeitet stattdessen auf der Ebene von Modellierungskonzepten wie Klasse, Attribut, Relation oder Objekt und kann deren Ausprägungen in Form alphanumerischer Listen/Tabellen darstellen (vgl. Abbildung 1.1). Diese Schnittstelle erlaubt jedoch einen modellübergreifenden Zugriff auf Informationen.

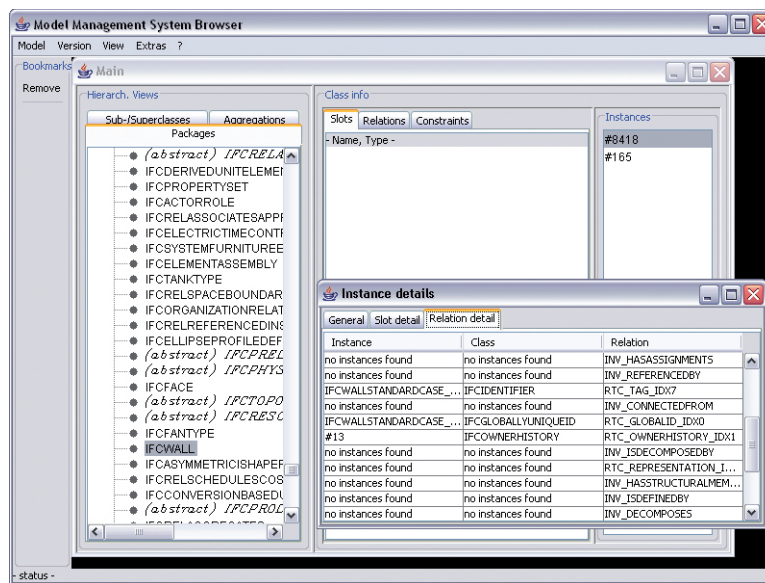


Abbildung 1.1.: Modellzugriff über den Browser der Modellierungsplattform

Das Finden des Planungskonzeptes muss als das Lösen eines mehr oder weniger komplexen Problems betrachtet werden, das oft nur vage umschrie-

Parallelität von Problemdefinition und Lösungsfindung

ben werden kann, bevor es gelöst ist [Ching, 1996]. Ziele und Anforderungen seitens des Bauherrn sind zu Projektbeginn oft unscharf formuliert. Detaillierte Kenntnisse bezüglich der zu integrierenden Bausubstanz fehlen. Eine wesentliche Aufgabe des Entwerfenden besteht somit in der Identifikation und Präzisierung von Anforderungen und deren Übertragung in konkrete Merkmale der Entwurfslösung [von Both, 2004]. Dies erfordert bei Projekten im Bestand sowohl den Dialog mit dem Bauherrn als auch den „Dialog“ mit der bestehenden Bausubstanz. Die Definition des Planungsziels ist somit ein integraler Teil der Lösung [Rowe, 1987], d. h. eine vollständige Vorformulierung bzw. eine für alle Arten von Projekten gültige Formalisierung des Lösungsweges ist nicht möglich.

*Notwendigkeit
modellübergreifenden
Zugriffs* Während der Entwicklung des Planungskonzeptes müssen verschiedene Aspekte der Bausubstanz einbezogen werden, wie beispielsweise Informationen zur Bauwerksgeometrie, zu Baumaterialien, zur Tragwerksstruktur oder zu bauphysikalischen Aspekten, die in der Regel auf die einzelnen fachspezifischen Modelle des Verbundes verteilt sind. Dies bedeutet, ein modellübergreifender Zugriff ist – vor allem in frühen Projektphasen – unbedingt notwendig.

*Orientierung in
räumlichen
Umgebungen* Menschen orientieren sich beim Umgang mit Bauwerken (Gebäuden) vorrangig an deren baulich-räumlichem Gefüge, d. h. einer konkreten und individuellen, dreidimensionalen, durchschreitbaren Struktur. Für solche Strukturen räumlicher Umgebungen verwenden sie anschauliche mentale Bilder, die als wesentliche Orientierungshilfen dienen [Lynch, 1965]. Architekten arbeiten beim Planen bzw. Entwerfen vorrangig mit zwei- und dreidimensionalen Darstellungen der Bauwerksgeometrie, wie beispielsweise Abbildung 1.2 zeigt. Diese können als Äquivalent zu den beim Entwerfen verwendeten mentalen Bildern betrachtet werden.

Problemsituation Hier ergibt sich eine Diskrepanz zwischen den bestehenden Möglichkeiten zum Zugriff auf bauwerksbezogene Informationen und der Denk- und

1.1. PROBLEMSTELLUNG

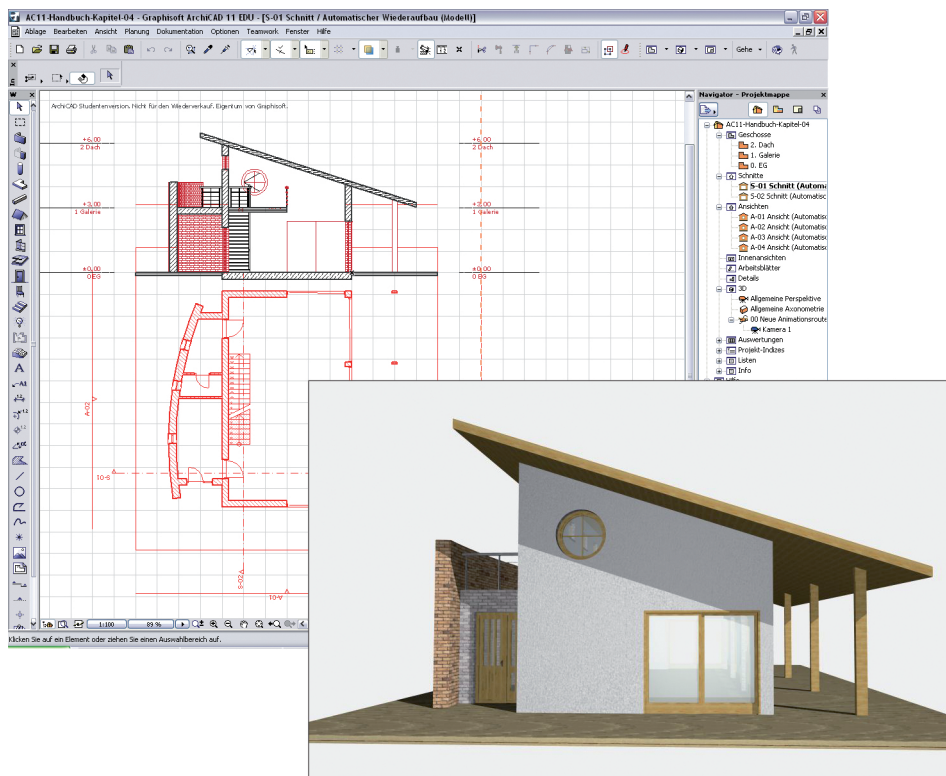


Abbildung 1.2.: Typische Arbeitsumgebung eines Architekten

Arbeitsweise der Entwerfenden bzw. Planenden. Die Struktur des Modellverbundes spiegelt anhand der beteiligten Modelle in erster Linie die Zusammensetzung eines Projektteams wider, nicht das Gefüge des betrachteten Bauwerks. Der modellübergreifende Zugriff auf Informationen ist nur auf einer wenig anschaulichen Ebene möglich, die vom Informationssuchenden einen hohen Aufwand zur Interpretation und Zuordnung der Informationen erfordert.¹ Dies stellt für einen Entwerfenden ein wesentliches Hindernis für einen intuitives Einbeziehen der verfügbaren Informationen in die Entwicklung eines Planungskonzeptes dar.

1.2. ZIELSETZUNG

Zwei betrachtete Ebenen Die Arbeit verfolgt zwei wesentliche Ziele. Auf der einen Seite stehen **Techniken zur Erschließung eines variablen Modellverbundes** für die Informationssuche im Kontext des Entwerfens, im Rahmen der Arbeit als *entwurfsbezogene Informationssuche* bezeichnet. Auf der anderen Seite soll ein Konzept für eine **Nutzerschnittstelle** entwickelt werden, die einem Entwerfenden eine flexible, intuitiv erkundende Suche im Datenbestand des Modellverbundes ermöglicht.

Erschließung des Modellverbundes Bei der Erschließung des Modellverbundes steht in erster Linie das Zusammenführen multipler, auf verschiedene Modelle verteilter Repräsentationen einzelner Bauwerkselemente im Mittelpunkt. Dafür ist einerseits eine den Modellverbund überlagernde Erschließungsstruktur zu entwickeln, die das baulich-räumliche Gefüge eines konkreten Bauwerks repräsentiert. Andererseits sind Verknüpfungen zwischen den Daten der einzelnen Modelle, d. h. den fachspezifischen Repräsentationen einzelner Bauwerkselemente, und den Elementen der Erschließungsstruktur zu definieren. Zur Verwaltung

¹ Der Leser überlege hierzu, wie sich in der generischen Modelldarstellung (Abbildung 1.1) die Suche nach der *nördlichen Aussenwand* gestaltet.

der Erschließungsstruktur und der Verknüpfungen wird eine entsprechende Datenstruktur benötigt, die in Form eines objektorientierten Schemas definiert werden soll.

Die Nutzerschnittstelle soll in erster Linie die Erschließungsstruktur für Informationssuchende explizit darstellen, wobei die Darstellungen mit den mentalen Bildern, die Menschen zur Orientierung im Bauwerk (Gebäude) verwenden, korrespondieren sollen. Auf Basis dieser Darstellung muss Informationssuchenden die Interaktion mit dem *erschlossenen Informationsraum* in der Form ermöglicht werden, dass Teilbereiche des Informationsraums gebildet und hinsichtlich relevanter Informationen untersucht werden können. Dabei ist zu berücksichtigen, dass Menschen verschiedene Vorgehensweisen bei der Informationssuche verwenden. Diese sind abhängig von der Art und Weise der Präsentation des Informationsraumes, seinen Strukturen sowie den durch die Schnittstelle unterstützten Interaktionsmöglichkeiten. Parallel ist die langfristig nicht absehbare Entwicklung im Bereich der Präsentations- und Interaktionsmöglichkeiten von Nutzerschnittstellen für elektronische Systeme zu berücksichtigen.

*Nutzerschnittstelle
zur
Informationssuche*

1.3. AUSGANGSPUNKTE UND VORGEHENSWEISE

Die Bereitstellung von Informationen muss stets kontextbezogen betrachtet werden, d. h. Informationen werden stets für einen bestimmten Zweck, eine bestimmte Nutzung und von einer bestimmten Nutzergruppe benötigt. Diese Aspekte haben entscheidenden Einfluss auf die Auswahl und Präsentation von Informationen. Daher ist es sinnvoll, für die Realisierung der genannten Zielstellung den Problembereich der Informationsbereitstellung für Planungsprozesse im Bauwerkslebenszyklus explizit abzugrenzen. Die für die vorliegende Arbeit zugrunde gelegte Situation ist durch die im Folgenden erläuterten Ausgangspunkte (AP) gekennzeichnet.

*Abgrenzung des
Problembereichs*

Gegenstand der zu entwickelnden Lösung ist der Umgang mit bestehender Bausubstanz, für den eine verbesserte Ausgangssituation durch die Archivierung und Bereitstellung digitaler Informationen zur Bausubstanz erreicht werden soll. Dabei ist der folgende Rahmen zu berücksichtigen:

Kenntnis der Bausubstanz *AP 1: Informationsbedarf* Planungsprojekte im Kontext bestehender Bausubstanz sind auf verlässliche Informationen zur bestehenden Substanz angewiesen. Ein für die gegebene Situation angemessenes Gesamtkonzept für ein solches Projekt kann nur entwickelt werden, wenn der Entwerfende über fundierte Kenntnisse zum bestehenden Bauwerk verfügt. Die frühen Projektphasen erfordern dabei besondere Sorgfalt, da die getroffenen Entscheidungen maßgeblich die Nachhaltigkeit des gesamten Projektes bestimmen. Vgl. dazu [Steinmann, 1997; Petzold, 2001]. Die Granularität des Informationsbedarfs verfeinert sich parallel zum Fortschreiten des Planungsstandes [Wiechmann, 1981].

Aufgrund der Vielfalt real existierender Bauwerkstypen und der z. T. erheblichen Unterschiede zwischen den einzelnen Stadien der Bauplanung sowie des Bauwerklebenszyklus hinsichtlich des Umgangs mit bauwerksbezogenen Informationen wird der betrachtete Kontext folgendermaßen eingeschränkt:

Frühe Planungsphasen *AP 2: Betrachteter Kontext* Primär werden die frühen Planungsphasen, d. h. das Finden des Planungskonzeptes durch den Architekten betrachtet. Von einer parallelen Aktivität anderer Fachplaner an der rechnerinternen Repräsentation des Bauwerks in diesen frühen Phasen wird abstrahiert. Das bedeutet, Modelländerungen durch parallel arbeitende Fachplaner und entsprechende Mechanismen der Kooperationsunterstützung werden nicht berücksichtigt, was jedoch die notwendige interpersonelle Kommunikation und Abstimmung zwischen Architekt und Fachplanern nicht ausschließt. Der Architekt wird jedoch als Einzel-

nutzer des zu entwickelnden Systems gesehen, für dessen Denk- und Arbeitsweise ein fachübergreifender Zugang zu den Daten des Modellverbundes zu realisieren ist. Zur Konzeptfindung in Planungsprojekten vgl. u. a. [Ching, 1996; Lorenz, 2004; Wiechmann, 1981].

Die Unterstützung der frühen Planungsphasen soll mit Hilfe computerbasierter Werkzeuge erfolgen, an die generell folgende Ansprüche zu stellen sind:

AP 3: Flexible Entwurfswerkzeuge Vor dem Hintergrund einer durchgängigen Projektbearbeitung, beginnend mit den frühen Entwurfsphasen, bestehen verschiedene Anforderungen an computerbasierte entwurfsunterstützende Werkzeuge. Die Werkzeuge sollten in Form einzelner Module realisiert sein, die entsprechend der zu lösenden Aufgabe vom Entwerfenden flexibel kombiniert und benutzt werden können. Dabei müssen die einzelnen Module eine durchgängige Bearbeitung des Entwurfsgegenstandes zulassen. Entwerfende müssen diese Werkzeuge intuitiv, d. h. entsprechend ihrer gewohnten Arbeitsweise, benutzen können. Dazu siehe u. a. [Donath, 2003; Steinmann, 1997; Petzold, 2001].

Ansprüche an Werkzeuge

Die betrachtete rechnerinterne Repräsentation eines Bauwerks legt die Verwendung objektorientierter Bauwerksmodelle zugrunde und kann folgendermaßen charakterisiert werden:

AP 4: Variabler Modellverbund Zur Verwaltung und Integration sowie lebenszyklusübergreifenden Bereitstellung bauwerksbezogener Informationen eignen sich Modellierungsansätze, die auf einem variablen Verbund fachspezifischer Modelle beruhen. Dazu siehe u. a. [Hauschild u. a., 2003; Willenbacher u. Hübler, 2004; Rüppel u. a., 2006]. Ein auf diese Weise verwalteter Datenbestand zu einem konkreten Bauwerk kann als Grundlage für die Planung von Instandhaltungs- oder

Datenhaltung

Modernisierungsmaßnahmen bzw. anderweitigen baulichen Veränderungen dienen.

Die gesuchte Nutzerschnittstelle fokussiert in erster Linie auf die Bereitstellung planungsrelevanter Informationen für die Tätigkeit eines Entwerfenden. Bei der Entwicklung von Werkzeugen, die kreative Schaffensprozesse unterstützen sollen, muss grundsätzlich von einer weiteren Annahme ausgegangen werden:

Kreative Schaffensprozesse *AP 5: Automatisierbarkeit* Entwurfsprozesse, zu denen die Entwicklung eines Gesamtkonzeptes für ein Projekt im Kontext bestehender Bausubstanz zu zählen ist, sind kreative Schaffensprozesse. Diese sind nicht bzw. nicht vollständig oder allgemeingültig formalisierbar und damit auch nicht automatisierbar. Vgl. dazu u. a. [Urchs, 2002] bzw. [Krech u. a., 1992, Bd. 4].

Entwerfen als Domäne des Menschen Dies bedeutet, das Finden eines angemessenen Gesamtkonzeptes für ein „Bestandsprojekt“ bleibt Domäne des Menschen. Dafür aber benötigen die Entwerfenden umfassende Informationen zu eben der bestehenden Bausubstanz, die den Ausgangspunkt des betreffenden Projektes darstellt. Im Zuge der Konzeptfindung entsteht Informationsbedarf in den meisten Fällen aber spontan und ist nicht vorhersehbar.

Bsp.: Installation von Aufzügen Angenommen, in einem mehrgeschossigen Wohngebäude sollen zusätzliche Personenaufzüge installiert werden. Verfolgt der Entwerfende das Ziel, die Aufzüge im Inneren des Gebäudes unterzubringen, werden insbesondere Informationen über die bestehende Tragstruktur benötigt. Strebt der Entwerfende dagegen für das gleiche Projekt eine Strukturierung der Fassade durch vorgelagerte, verglaste Aufzugsschächte an, werden vor allem Informationen zur Lage des Gebäudes auf dem Grundstück benötigt. Denn in diesem Fall muss u. a. die Einhaltung von Abstandsflächen berücksichtigt werden.

Folglich ist eine Automatisierung der Auswahl von Informationen aus dem Datenbestand nicht möglich. Ebenso wenig kann die Art und Weise der Präsentation der Informationen vorab festgelegt werden bzw. automatisch erfolgen.

Aus den Punkten 1 bis 5 folgend wird für die Arbeit folgende Ausgangssituation angenommen: Für ein bestehendes Bauwerk sei eine Dokumentation verfügbar, deren Datenbestand in Form eines variablen Verbundes objektorientierter fachspezifischer Bauwerksmodelle verwaltet wird. Um die verwalteten Informationen für die Planung, insbesondere die frühen Phasen, verfügbar zu machen, wird eine Nutzerschnittstelle für die Informationssuche in der Bauwerksdokumentation benötigt. Diese muss „intuitiv benutzbar“ sein und den Entwerfenden erlauben, flexibel nach „spontan benötigten“ Informationen zu suchen. Die Gesamtheit aus Datenbestand und Nutzerschnittstelle für die Informationssuche soll im Folgenden allgemein als *Informationsumgebung* bezeichnet werden.

Ausgangssituation

Die Lösung des so umrissenen Problems erfordert zunächst eine Präzisierung der Zielkriterien, die bei der Entwicklung der benötigten Nutzerschnittstelle zu berücksichtigen sind. Vor diesem Hintergrund ergeben sich folgende Fragen:

Frage a): Welchen Ansprüchen müssen „intuitiv benutzbare“ Werkzeuge für die entwurfsbezogene Informationssuche genügen? Was genau bedeutet „intuitive Benutzbarkeit“ im konkreten Fall der gesuchten Nutzerschnittstelle?

*Intuitive
Benutzbarkeit*

Grundsätzlich ist intuitive Benutzbarkeit dann gegeben, wenn Benutzer eines Werkzeugs es entsprechend gewohnter Arbeitsweisen verwenden können, bzw. wenn das Werkzeug in einer Form selbsterklärend ist, dass es den Lernaufwand zur Benutzung minimiert. Vor diesem Hintergrund ist eine Betrachtung sowohl der Vorgehensweisen beim Entwerfen als auch bei der

Informationssuche erforderlich. Dabei sollen folgende, sich aus der Frage a) ableitende Fragestellungen zur Orientierung dienen:

Entwerfen als Verstandesleistung *Frage b):* Wodurch wird die Leistungsfähigkeit eines Entwerfenden beeinflusst? An diese Frage knüpfen weitere an, wie z.B.: Wie gehen Entwerfende bei der Entwicklung einer Entwurfslösung vor? Welche Leistungen müssen sie erbringen? Wie werden diese Leistungen erbracht?

Ziel dieses Fragekomplexes ist die Präzisierung der bisher genannten Anforderungen an entwurfsunterstützende Werkzeuge. Im Bereich der Architektur ist zu diesem Thema nur wenig Literatur verfügbar [Joedicke, 1976; Rowe, 1987; Gero, 1999; Lorenz, 2004]. Aus diesem Grund wird zusätzlich Material aus der Kognitionspsychologie herangezogen, das eine Betrachtung des Entwerfens aus der Sicht des kreativen Problemlösens ermöglicht.

Parallel zum Entwerfen sind Randbedingungen und Vorgehensweisen bei der Informationssuche zu berücksichtigen. Somit ergibt sich ein weiterer Fragekomplex:

Informationssuche als Verstandesleistung *Frage c):* Welche Faktoren beeinflussen Menschen bei der Informationssuche? Damit hängen Fragen zusammen, wie u. a.: Wie gehen Menschen bei der Suche nach Informationen generell vor? Wodurch wird der Erfolg der Suche, vor allem in unbekanntem Informationsumgebungen, beeinflusst? Welche Auswirkungen hat die Klarheit, mit der ein Informationsdefizit formuliert werden kann, auf die Vorgehensweise bei der Suche? Welche Wechselwirkungen bestehen zwischen der Beschaffenheit einer Informationsumgebung und den Vorgehensweisen bei der Suche?

Mit Hilfe dieses Fragekomplexes sollen die Anforderungen an Werkzeuge zur (entwurfsbezogenen) Informationssuche präzisiert werden. Zu diesem Zweck ist eine Betrachtung des Prozesses der Informationssuche als Verstandesleistung erforderlich. Sowohl aus der Psychologie als auch aus dem

Informations- und Dokumentationswesen ist Material zu dieser Problematik verfügbar, das entsprechend der Fragestellungen aufgearbeitet werden soll.

Auf Basis der gefundenen Antworten sowie unter Rückgriff auf zwei Analogien aus dem Bereich des Informations- und Dokumentationswesens wird im Anschluss ein Gesamtkonzept für die zu realisierende Informationsumgebung formuliert. Die zentralen Komponenten dieser Informationsumgebung, die den bestehenden Modellverbund ergänzen, werden schließlich unter Verwendung einer standardisierten grafischen Notation beschrieben und in Teilen exemplarisch realisiert.

*Gesamtkonzept einer
Informationsumgebung*

1.4. AUFBAU DER ARBEIT

Insgesamt gliedert sich die Arbeit in vier Stufen:

- Stufe I: Grundlagen
- Stufe II: Allgemeine Lösung
- Stufe III: Funktionale und spezifische Lösung
- Stufe IV: Evaluierung.

Abbildung 1.3 zeigt schematisch die inhaltliche Struktur der Arbeit.

STUFE I - GRUNDLAGEN

Die folgenden Kapitel 2, 3, 4 und 5 behandeln die Themenkomplexe, die das theoretische Fundament der Arbeit bilden.

Kapitel 2 untersucht allgemein das Planen und Bauen im Kontext bestehender Bausubstanz, denn dieses stellt das spätere Nutzungsumfeld der zu entwickelnden Informationsumgebung dar. **Kapitel 3** fasst den derzeitigen

Bauwerksmodellierung

1. EINFÜHRUNG

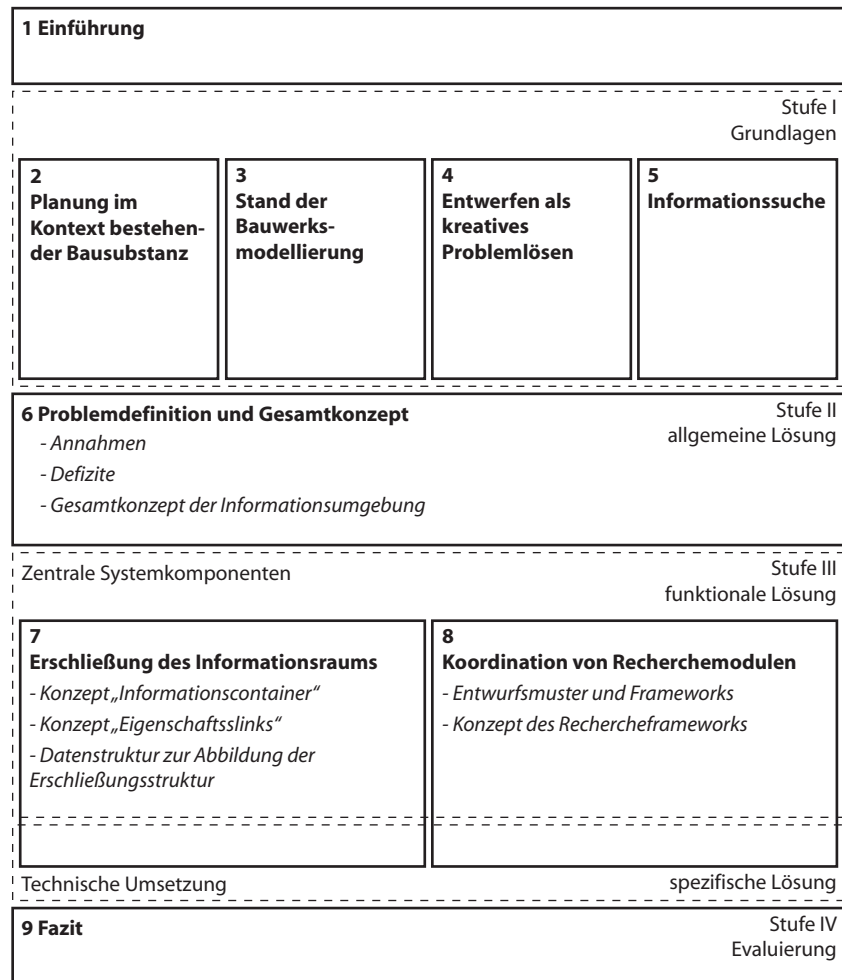


Abbildung 1.3.: Aufbau der Arbeit

Kenntnisstand in der Bauwerksmodellierung in einem Überblick zusammen, wobei in erster Linie die technischen Möglichkeiten zur Verwaltung einer digitalen Bauwerksdokumentation erfasst werden sollen. Diese beiden Kapitel dienen vor allem der Bestandsaufnahme der gegebenen Ausgangssituation.

Die zwei folgenden Kapitel widmen sich in erster Linie der Beantwortung der genannten Fragekomplexe. Schwerpunkt des **Kapitels 4** ist dabei die vertiefende Betrachtung des Entwerfens als menschliche Verstandesleistung. In diesem Zusammenhang steht die Beantwortung der Fragen b) zum Entwerfen. **Kapitel 5** setzt sich mit der Problematik Informationssuche auseinander und zielt auf die Beantwortung des Fragekomplexes c). Dabei steht vor allem die Informationssuche in digital repräsentierten Informationsräumen im Mittelpunkt. Denn als solcher muss der als gegeben angenommene Datenbestand betrachtet werden.

Entwerfen

*Informationssuche/
Informations-
strukturierung*

STUFE II - ALLGEMEINE LÖSUNG

Kapitel 6 führt die Erkenntnisse der Grundlagenuntersuchungen zusammen, wobei zunächst der übergreifende Fragekomplex a) zur intuitiven Benutzbarkeit zusammenfassend beantwortet wird. Ausgehend von den so konkretisierten Anforderungen werden die zu überwindenden Defizite identifiziert bzw. präzisiert. Daran anknüpfend wird ein Gesamtkonzept für eine Informationsumgebung formuliert. Diese Systemphilosophie bildet den Ausgangspunkt für die Konkretisierung der zentralen Systemkomponenten der gesuchten Informationsumgebung, der sich die beiden folgenden Kapitel widmen.

*Formulierung des
Gesamtkonzeptes*

STUFE III - FUNKTIONALE LÖSUNG

- Konkretisierung wesentlicher Systemkomponenten* Ausgehend vom entwickelten Gesamtkonzept wird eine funktionale Lösung für die Informationsumgebung entwickelt. Im Mittelpunkt steht dabei einerseits eine für die Informationssuche sinnvolle Strukturierung des Informationsraums, andererseits die Entwicklung einer Nutzerschnittstelle, die Informationssuchenden die Interaktion mit dem digital repräsentierten Informationsraum ermöglicht.
- Erschließungsstruktur* **Kapitel 7** widmet sich der Erschließung der in einem variablen Modellverbund verwalteten Informationen zu einem Bauwerk aus der Sicht des Entwerfenden. Dabei steht das baulich-räumliche Gefüge des konkreten Bauwerks als oberstes Strukturierungsparadigma im Vordergrund. Zur Verwaltung entsprechender Erschließungsstrukturen wird eine Datenstruktur entwickelt und unter Verwendung der Unified Modelling Language (UML) auf einer implementierungsunabhängigen Ebene beschrieben.
- Framework als Basis einer flexiblen Nutzerschnittstelle* **Kapitel 8** setzt sich mit der Entwicklung der Nutzerschnittstelle für die Informationsumgebung auseinander. Aufgrund der Flexibilität, die eine solche Schnittstelle bieten muss, ist eine Realisierung als ein monolithisches Softwaresystem nicht sinnvoll. Daher wird ein System austauschbarer Präsentations- und Interaktionsmodule vorgeschlagen, die auf der Basis eines gemeinsam genutzte Grundfunktionalitäten bereitstellenden Frameworks integriert und koordiniert werden. Schwerpunkt des Kapitels ist die Entwicklung des Frameworks als ein objektorientiertes System sowie dessen Beschreibung unter Verwendung der UML.

STUFE III - SPEZIFISCHE LÖSUNG

In den Kapiteln 7 und 8 werden abschließend jeweils technische Möglichkeiten zur Implementierung der vorgeschlagenen funktionalen Lösungen diskutiert. Eine vollständige Implementierung der Informationsumgebung als ein Gesamtsystem ist jedoch nicht Gegenstand der Arbeit.

*Vorschläge zur
technischen
Umsetzung*

STUFE IV - EVALUIERUNG

Kapitel 9 fasst schließlich die Erkenntnisse der Arbeit zusammen. Gleichzeitig wird der vorgeschlagene Lösungsansatz kritisch betrachtet und hinterfragt. Dies betrifft in erster Linie die Diskussion des für die Lösungsfindung gesetzten Kontext *Finden des Planungskonzeptes*, d. h. der in diesem Zusammenhang angenommenen Randbedingungen und deren Einfluss auf die gefundene Lösung. Dabei soll vor allem diskutiert werden, ob und mit welchen Konsequenzen der Lösungsansatz auf einen erweiterten Kontext übertragen werden kann. Dies betrifft sowohl andere Projektsituationen, wie beispielsweise die Neubauplanung oder spätere Projektphasen, in denen etwa die aktive Kooperation verschiedener Planungsbeteiligter über die rechenrinterne Repräsentation des Bauwerks zu berücksichtigen ist. Ebenso ist eine Übertragbarkeit auf andere Bauwerkslebensphasen, insbesondere die Bauwerksnutzung und -verwaltung, zu diskutieren.

*Wertung der
Ergebnisse*

1. EINFÜHRUNG

2. PLANUNG IM KONTEXT BESTEHENDER BAUSUBSTANZ

Die Erhaltung, Instandsetzung und Modernisierung bestehender Bausubstanz hat einen nicht zu vernachlässigenden Anteil am Gesamtauftragsvolumen des Baugewerbes. Europaweit finden etwa 40% aller Planungstätigkeiten im Kontext bestehender Bausubstanz statt [Maisberger Whiteoaks/-Nemetschek, 2005]. Knapp 70% der in der genannten Studie befragten deutschen Architektur- und Planungsbüros nannten das Bauen im Bestand als ein wesentliches Segment für zukünftige Betätigungsfelder.

Ausgangspunkt

Nach einer Studie von Hommerich waren in allen deutschen Bundesländern zwischen 1997 und 2001 sinkende Neubautätigkeiten und gleichzeitig ein steigender Anteil an Instandhaltungsmaßnahmen zu verzeichnen [Hommerich u. a., 2005]. Etwa 40% aller Investitionen auf dem Immobilienmarkt in der Bundesrepublik Deutschland floss in Modernisierungsmaßnahmen und nur etwas mehr als 20% in Neubauten einschließlich des Erwerbs von Bauland. Vgl. dazu auch Abbildung 2.1.

Die Planung von Baumaßnahmen im Kontext bestehender Bausubstanz soll im Folgenden kurz als *Planung im Bestand* bezeichnet werden. Dieses Kapitel erläutert Besonderheiten und wesentliche Randbedingungen der Planung im Bestand.

Inhalt

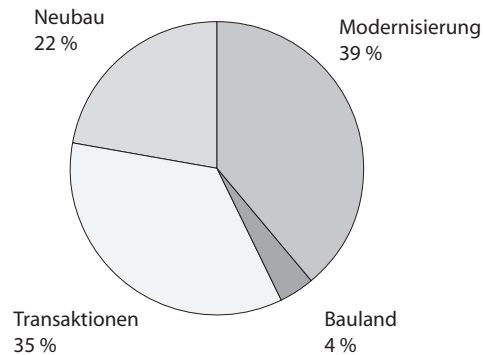


Abbildung 2.1.: Investitionen am Immobilienmarkt 2003, nach [Hommerich u. a., 2005, S. 31]

2.1. BEGRIFFE

Bauwerk Unter einem **Bauwerk** versteht man im Allgemeinen eine dauerhafte Baukonstruktion, die an einen festen Standort gebunden ist und mit Mitteln der Bautechnik errichtet wurde. Zu Bauwerken zählen nicht nur Leistungen des Hochbaus, sondern auch Ingenieurbauten und Leistungen des Tiefbaus. Zur Definition des Begriffs vgl. u. a. [Kadatz, 1994; Koepf, 1999].

Gebäude **Gebäude** sind nach [Kadatz, 1994] Bauwerke, die eine größtenteils überdachte, in sich geschlossene Raumordnung bilden. Sie dienen im Wesentlichen als bauliche Hülle für menschliche Tätigkeiten. Gebäude sind in der Regel oberirdisch bzw. erheben sich über die Erde [Koepf, 1999].

Die im Rahmen dieser Arbeit betrachtete Problematik des Planens und Bauens im Bestand bezieht sich vor allem auf Gebäude, d. h. auf die Umnutzung, den Umbau oder die Erweiterung einer bestehenden baulichen Hülle bzw. deren Ausbau. Ziel ist dabei die Anpassung der bestehenden Bausubstanz an veränderte Ansprüche der darin stattfindenden menschlichen Aktivitäten.

Bauwerke allgemein werden dann als **Architektur** bzw. architektonische Leistung angesehen, wenn der Bauwille über die reine Notwendigkeit bzw. reine Nützlichkeitsabwägungen hinaus geht [Koeopf, 1999]. Diese Umschreibung verweist bereits darauf, dass die Bewertung von Architektur einer Reihe von Kriterien unterliegt, die Wertvorstellungen beinhalten und somit nicht objektiv quantifizierbar sind. Diese Kriterien beeinflussen maßgeblich auch den Prozess des Entwerfens von Architektur bzw. der Planung baulicher Veränderungen an bestehenden architektonischen Objekten.

Architektur

Unter **Planung** wird im Allgemeinen die Bearbeitung der Leistungsphasen 1 (Grundlagenermittlung) bis 7 (Mitwirkung bei der Vergabe) der Honorarordnung für Architekten und Ingenieure [HOAI, 1996] verstanden. Der Begriff **Entwerfen** bezeichnet dabei das Finden des Planungskonzeptes in den frühen Projektphasen (LP. 1-3, HOAI), was einen kreativen Schaffensprozess darstellt. Der Fokus der vorliegenden Arbeit liegt explizit auf der Unterstützung der frühen Planungsphasen von Projekten im Kontext bestehender Bausubstanz. Für diese Phasen muss von einem hohen Anteil kreativer Tätigkeit im Sinne des Entwerfens ausgegangen werden. Vor diesem Hintergrund wird in der Arbeit explizit *Entwerfen* anstelle von *Planen* verwendet.

*Planen und
Entwerfen*

2.2. NACHHALTIGES PLANEN

Nachhaltigkeit als eine dauerhafte, auf künftige Generationen ausgerichtete, evolutionär sinnvolle Entwicklungsstrategie, die ein gesundes und produktives Leben im Einklang mit der natürlichen Umwelt in den Mittelpunkt stellt, rückt immer stärker in den gesellschaftlichen Fokus. Elementares Prinzip nachhaltiger Entwicklungen ist die deutliche Reduzierung der Energie- und Stoffumsätze zur Erledigung einer bestimmten Aufgabe. Dazu vgl. [Vester, 1999; Pfeiffer u. a., 2001].

Nachhaltiges Planen

- Erhaltung statt
Neubau* Unter dem Gesichtspunkt des nachhaltigen Umgangs mit bestehenden Ressourcen sollte der Erhaltung bestehender Bausubstanz und ggf. deren Anpassung an veränderte Anforderungen grundsätzlich der Vorzug gegenüber dem Abriss und Neubau gegeben werden. Unter diesem Anspruch sind bauliche Veränderungen als ein integraler Bestandteil des Lebenszyklus von Bauwerken zu betrachten, sei es zur Instandhaltung, zur Modernisierung oder zur Anpassung der bestehenden Substanz an anderweitig veränderte Nutzungsanforderungen.
- Entscheidungen
früher
Planungsphasen
bestimmen* Unter dem Blickwinkel der langen Lebens- und Nutzungsdauer von Bauwerken liegt ein enormes Potential für die Entwicklung nachhaltiger Lösungen in den frühen Projektphasen, in denen der Rahmen für das Projekt und sämtliche damit verbundenen Maßnahmen abgesteckt werden. Die frühen Planungsphasen beeinflussen sowohl die Kosten und die Qualität des Bauwerks, als auch die Kosten für die Nutzung, den Betrieb und die Instandhaltung des Bauwerks. Etwa 90% der Baukosten werden durch Entscheidungen in diesen Phasen determiniert. Bis zu diesem Zeitpunkt sind jedoch nur etwa 2% der Baukosten aufgrund der Planungsleistungen tatsächlich angefallen (vgl. Abbildung 2.2 sowie [Steinmann, 1997]). Bezogen auf den Nutzungszeitraum ist die Bedeutung der frühen Planungsentscheidungen als noch höher einzuschätzen. Gemessen am Entscheidungsspielraum des Planers haben Nutzer oder Betreiber eines Gebäudes nur noch bedingt Einfluss auf die Kosten der Bauunterhaltung und die damit verbundenen Stoff- und Energieströme [Pfeiffer u. a., 2001].
- Nachhaltigkeit sowie
Kosten*
- Kenntnis der
Bausubstanz als
Voraussetzung* Das Finden des Planungskonzeptes für ein Projekt im Bestand muss grundsätzlich von der bestehenden Bausubstanz ausgehen und diese als einen integralen Bestandteil des Konzeptes einbeziehen. Ein angemessenes und nachhaltiges Entwurfskonzept kann damit nur auf der Basis fundierter Kenntnisse der bestehenden Substanz entstehen. Die Auseinandersetzung mit dem Bestand und dessen eingehende Analyse bildet daher einen we-

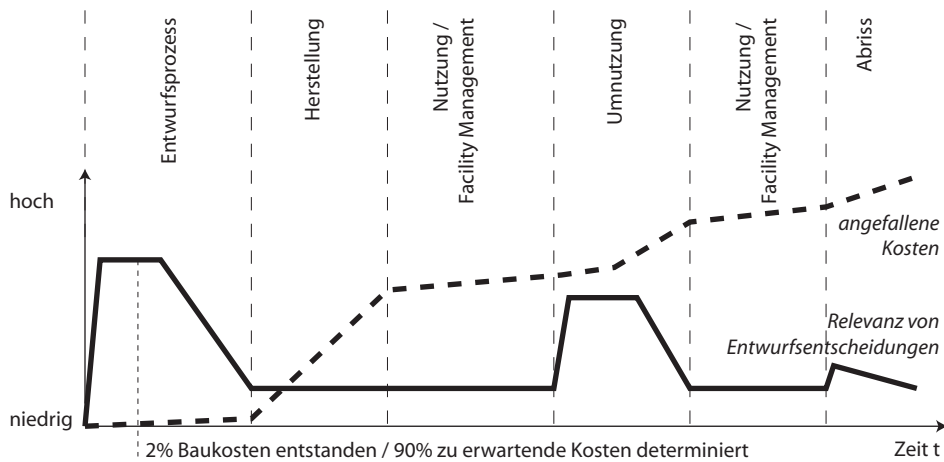


Abbildung 2.2.: Relevanz von Entwurfsentscheidungen früher Planungsphasen nach [Steinmann, 1997, S. 5]

sentlichen Anteil der Planungstätigkeit.

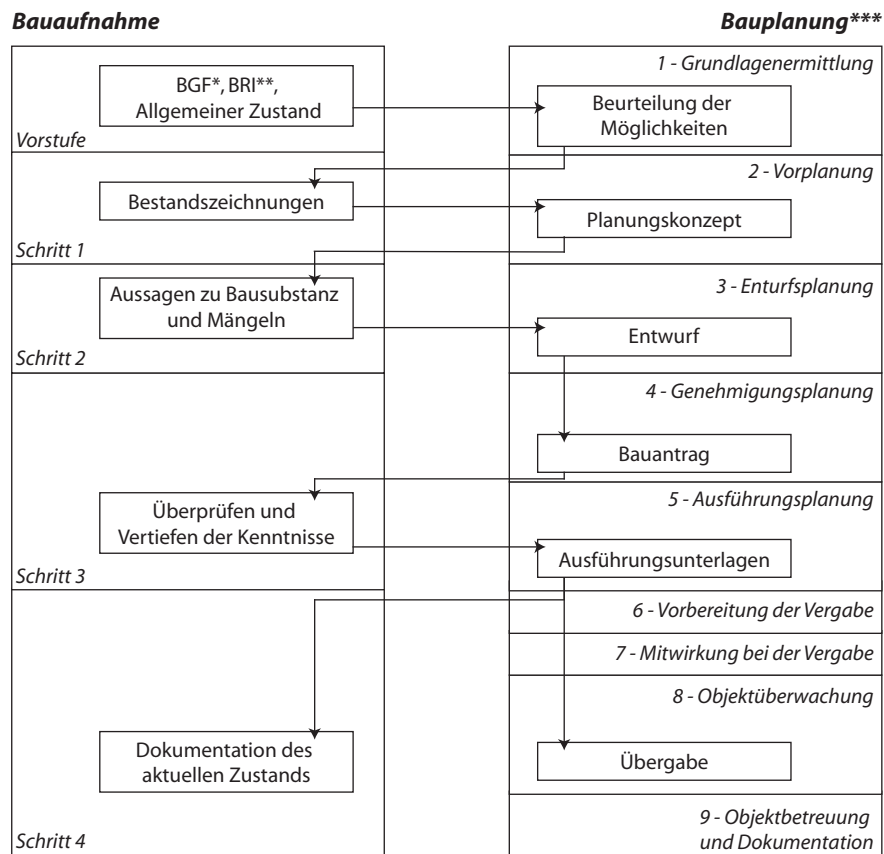
Dabei erfolgt parallel zur Konkretisierung des Entwurfskonzeptes eine schrittweise Annäherung an das bestehende Bauwerk. Wiechmann schlägt dafür eine Vorgehensweise in fünf Stufen vor, bei der parallel zum Planungsfortschritt der Kenntnisstand bezüglich der bestehenden Substanz vertieft und erweitert wird [Wiechmann, 1981]. Dieser Ablauf korrespondiert mit den in der HOAI beschriebenen neun Leistungsphasen (siehe Abbildung 2.3 sowie [HOAI, 1996]).

*Parallelität von
Planungsfortschritt
und
Informationsbedarf*

Daraus wird deutlich, dass über den gesamten Verlauf der Planung immer wieder eine Rückkopplung der Planungsentscheidungen mit Informationen zum bestehenden Bauwerk stattfindet. Vor allem in den Stufen 1 und 2, in denen das globale Planungskonzept für ein Projekt erarbeitet wird, ist die Verfügbarkeit aller für das bestehende Bauwerk charakteristischen wie auch kritischen Informationen eine unbedingte Notwendigkeit.

*Permanente
Rückkopplung*

2. PLANUNG IM KONTEXT BESTEHENDER BAUSUBSTANZ



* BGF - Brutto-Grundfläche
 ** BRI - Brutto-Rauminhalt
 *** Leistungsphasen nach [HOAI, 1996]

Abbildung 2.3.: Vorgehensweise bei Modernisierungsprojekten nach [Wiechmann, 1981, S. 97]

2.3. BEREITSTELLUNG PLANUNGSRELEVANTER INFORMATIONEN

In den meisten Fällen stehen Bestandsunterlagen nicht oder nur in mangelhafter Qualität zur Verfügung. Daher dient üblicherweise die Bauaufnahme der Erfassung und Bereitstellung der für die Planung relevanten Informationen zum bestehenden Bauwerk. Der von Wiechmann vorgeschlagene Ablauf sieht dabei eine zur Planung parallel stattfindende Bauaufnahme vor. Während der fünf Stufen werden die erfassten Informationen dabei schrittweise verfeinert. Insbesondere in Bereichen, in denen verändernde Maßnahmen an der Bausubstanz vorgesehen sind, ist eine intensive Auseinandersetzung mit der Substanz notwendig. Von geplanten Eingriffen nicht betroffene Bauwerksteile müssen dagegen weniger detailliert erfasst werden.

*Planungsbegleitende
Bauaufnahme*

Im Gesamtvolumen von Altbauprojekten wird der zur Erfassung bauwerksbezogener Informationen erforderliche Aufwand im Allgemeinen als feste Position einkalkuliert [Wiechmann, 1981; HOAI, 1996; Schmitz u. Krings, 2004]. Somit entstehen in Abhängigkeit von Umfang und Detaillierungsgrad der zu erfassenden Informationen zusätzliche Baunebenkosten (vgl. Tabelle 2.1). Dies wird in der Regel in Kauf genommen, denn unvollständige oder fehlerhafte Informationen sind eine potentielle Quelle für Fehlplanungen und daraus entstehende Mehraufwendungen [Wiechmann, 1981; GISMO Projekt, 2008].

*Kosten für die
Bauaufnahme*

Die während der Bauaufnahme zu erfassenden und zu dokumentierenden Informationen können nach [Cramer, 1993; Wangerin, 1992] drei Gruppen zugeordnet werden:

*Zu erfassende
Informationen*

- Bauwerksgeometrie
- Baumaterialien und konstruktive Zusammenhänge
- vorgefundener Zustand des Bauwerks.

2. PLANUNG IM KONTEXT BESTEHENDER BAUSUBSTANZ

Anforderung / Leistung (Kosten in EUR/m ² BGF)	Einfach z.B. größere Wohngebäude, Wohnanlagen mit Wiederho- lungen	Normal z.B. Einzelge- bäude mit 5-15 WE, „typischer Altbau“	Gehoben z.B. kleinere Einzelgebäude, technische Be- sonderheiten, Sonderunter- suchungen
<i>Maßliche Bestandsaufnahme</i>	2,50	3,00	3,50
<i>Erstellung Bestandspläne</i>	2,50	3,00	3,50
<i>Technische Bestandsaufnah- me</i>	3,00	5,00	7,00

Tabelle 2.1.: Kosten für die Bauaufnahme nach [Schmitz u. Krings, 2004, S. 35]

Bauaufnahme als Umkehrung des Entstehungsprozesses Die beiden genannten Autoren sehen dabei den Bauaufnahmeprozess als gedanklich vollzogene Umkehrung des Entstehungs- und Entwicklungsprozesses des Bauwerks, der die vorgefundene Bausubstanz wieder in Pläne (rück-)überführt. Während dieses gedanklichen Prozesses muss die Lebensgeschichte des Bauwerks vom Entwurf über die Planung und Bauausführung und die späteren baulichen Veränderungen bis hin zum gegenwärtigen Zustand nachvollzogen werden.

Rekonstruktion von Bestandsinformationen Ein für die Bauaufnahme grundsätzliches Problem liegt vor allem darin, dass am real existierenden Bauwerk zunächst ausschließlich die Oberflächen der raumbegrenzenden Elemente wahrgenommen werden können. Das bedeutet, dass zwar die Bauwerksgeometrie mit hoher Sicherheit erfasst werden kann, jedoch bezüglich Baumaterialien, konstruktiver Zusammenhänge und Zustand des Bauwerks zunächst nur Vermutungen angestellt werden können. Dennoch werden diese Informationen in verlässlicher Form benötigt und müssen oft in einem zeitaufwändigen Prozess rekonstruiert werden.

Vermutungen aufgrund von Erfahrung Das konstruktive Gefüge des Bauwerks ist in der Regel durch Oberflächenverkleidungen verdeckt. Durch Vergleiche mit ähnlichen Bauwerken

können verwendete Materialien und Konstruktionen zwar vermutet werden, diese Vermutungen sind aber im Fall eines Eingriffs in die Substanz vor Ort zu prüfen. Bauliche Schäden, vor allem an konstruktiven Elementen sind zudem nicht immer offensichtlich. Sind konstruktive Zusammenhänge bekannt, können Planer aufgrund von Erfahrungen auf typische Schadensbilder schließen und ggf. überprüfende Untersuchungen veranlassen. Siehe dazu [Wiechmann, 1981; Petzold, 2001; Thurow, 2004].

DURCHGÄNGIGER INFORMATIONSFLUSS IM BAUWERKSLEBENSZYKLUS

Zur Herstellung des Gebäudes waren jedoch vielfältige Informationen zu Materialien und zum konstruktiven Gefüge erforderlich und wurden somit im Zuge der Planung detailliert dokumentiert.

*Planungsunterlagen
für Neubau*

Planungsunterlagen werden heute meist mit Hilfe computergestützter Systeme erstellt. Folglich kann davon ausgegangen werden, dass nach Abschluss der Planung ein rechnerinternes Abbild des Bauwerkes existiert, dessen Beschaffenheit an dieser Stelle noch nicht weiter hinterfragt werden soll. Die Problematik der Datenverwaltung im Bauwesen wird im folgenden Kapitel detaillierter betrachtet.

Selbst unter Berücksichtigung von Änderungen während der baulichen Ausführung kann weiterhin davon ausgegangen werden, dass die in den Entwurfsphasen entwickelte Struktur des Gebäudes über seinen gesamten Lebenszyklus hinweg (in den wesentlichen Teilen) erhalten bleibt. Somit ist die Forderung nach der Erhaltung des während der Planung erstellten Abbilds – und gegebenenfalls dessen Anpassung an den gebauten Zustand – die logische Konsequenz.

*Erhaltung des
rechnerinternen
Abbilds...*

Die Forderung nach einer lebenszyklusübergreifenden Bauwerksdokumentation wurde bereits von anderen Autoren formuliert [Gessmann, 2005; Steinmann, 1997]. In [GISMO Projekt, 2008] wird ausdrücklich der ver-

*...als digitale
Bauwerksakte*

stärkte Wunsch seitens der Bauherren nach einer digitalen Bauakte hervor-
gehoben. Diese soll einerseits den Entwurfsprozess und die anschließenden
Bauphasen mit allen Änderungen nachvollziehbar machen, andererseits als
Datengrundlage für die Verwaltung und den Betrieb des Bauwerks dienen.

*Kein durchgängiger
Informationsfluss*

Praktisch sind jedoch insbesondere am Übergang von der Bauplanung/
Bauausführung zur Gebäudenutzung Brüche im Informationsfluss und In-
formationsverluste zu verzeichnen [Gessmann, 2005], wie Abbildung 2.4
zeigt. Dies mag darauf zurückzuführen sein, dass ein großer Teil der zur
Herstellung erforderlichen Daten für die Verwaltung des Gebäudes nicht
unmittelbar relevant ist und somit nicht überführt wird oder in verfügbare
Anwendungen nicht überführt werden kann.

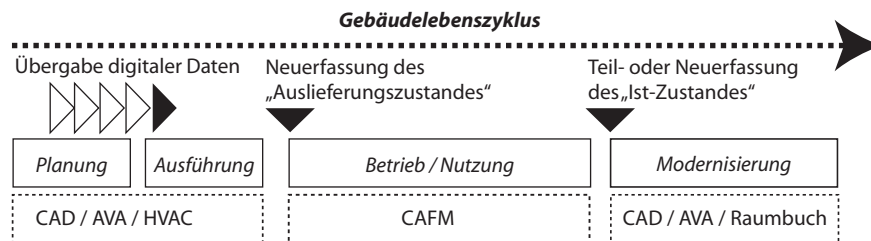


Abbildung 2.4.: Brüche im Informationsfluss nach [Gessmann, 2005, S. 2]

*Fortschreibung der
Daten*

Dennoch hat der während der Planung erstellte Datensatz einen nicht
zu vernachlässigenden Wert. Die potentielle Relevanz für spätere bauliche
Veränderungen, Instandhaltungs- oder Modernisierungsmaßnahmen spricht
zudem für die lebenszyklusübergreifende Erhaltung der Daten und deren
Fortschreibung und Abgleich in Korrespondenz zum gebauten Zustand.

2.4. SPEZIALISIERUNG UND ROLLENVERTEILUNG IM PLANUNGSPROZESS

Aktuelle Bauaufgaben haben einen Grad an Komplexität erreicht, der durch eine einzelne Person nach dem Vorbild des mittelalterlichen Baumeisters nicht mehr sinnvoll beherrscht werden kann. Die Bauplanung erfolgt heute in der Regel in einem arbeitsteiligen Prozess, bei dem einzelne Planungsleistungen von spezialisierten Fachplanern wahrgenommen werden. Dennoch obliegt das Finden des Planungskonzeptes beziehungsweise die Entscheidung darüber, nach wie vor Einzelnen. Lorenz belegt anhand von Befragungen, dass Entwerfen im Team durchaus Vorteile hat, jedoch einer klaren Regieführung und Hierarchie bei der Entscheidungsfindung bedarf [Lorenz, 2004]. Vester, aber auch von Both betonen ausdrücklich, dass bei der Planung in komplexen, offenen Systemen eine Zerlegung des Gesamtsystems in einzelne Teilsichten sowie die separate Bearbeitung der Teilsichten nicht zielführend ist, solange das Ganze nicht gleichzeitig in seinem Gesamtzusammenhang betrachtet wird [Vester, 1999; von Both, 2004]. Dabei müssen Wechselwirkungen über die Grenzen der Teilsichten hinaus berücksichtigt werden, um unerwünschte Nebeneffekte auszuschließen.

Komplexität und Arbeitsteilung

Projekt im Gesamtzusammenhang betrachten!

Zudem ist eine Vielzahl von Aspekten beim Entwurf von Architektur, unabhängig davon, ob es sich um eine Neuentwicklung oder eine Umgestaltung bestehender baulicher Situationen handelt, nicht objektiv quantifizierbar. Dazu zählen beispielsweise ästhetische Ansprüche, aber auch Vorlieben und Gewohnheiten der tatsächlichen oder potentiellen Nutzer eines Gebäudes. Diese Aspekte sind nicht Bestandteil digitaler Bauwerksmodelle, sondern werden ausschließlich durch den Entwerfenden in die Lösungsfindung einbezogen.

Wertvorstellungen

*Architekt in
federführender Rolle*

Daher ist es notwendig, dass eine Person oder eine sehr kleine Gruppe von Personen innerhalb des Projektteams die Verantwortung für den Gesamtzusammenhang des Projektkonzeptes übernimmt bzw. trägt. Im Sinne des klassischen Berufsbildes fällt diese Rolle in der Regel dem Architekten zu. Inhalt dieser Rolle ist einerseits das Schaffen eines vermittelnden Ausgleichs zwischen den Aspekten *Funktion, Form und Gestalt* sowie *Konstruktion*. Andererseits zählt zu den Architektenaufgaben ebenso die Koordination und Integration einzelner Fachplanerleistungen. Die Architektenrolle ist somit als die eines Generalisten zu sehen, bei dem die einzelnen Fäden der Planung quasi zusammenlaufen. Dies bedeutet, dass der Architekt – zumindest in den Grundzügen – über alle Aspekte des Projektes informiert sein muss. Nur so ist es möglich, das Projekt – beginnend mit der Phase der Konzeptfindung – als Ganzes sinnvoll zu koordinieren.

2.5. COMPUTERGESTÜTZTE WERKZEUGE FÜR DIE PLANUNG IM BESTAND

*Durchgängigkeit ist
nicht gegeben*

Eine durchgehende Bearbeitung von Planungsprojekten im Kontext bestehender Bausubstanz auf der Basis computergestützter Werkzeuge ist gegenwärtig nicht realisiert. Die Entwicklung entsprechender Werkzeuge stellt einen aktuellen Forschungsschwerpunkt dar [Donath u. Petzold, 2005; GIS-MO Projekt, 2008; Pfeiffer u. a., 2001].

*Prototypen für die
Planung im Bestand*

Verfügbare kommerzielle wie auch experimentelle Werkzeuge zur Unterstützung des architektonischen Entwurfs- und Planungsprozesses fokussieren fast ausschließlich auf die Neubauplanung [Donath u. Petzold, 2005]. Für ausgewählte Aspekte der Planung im Bestand wurden in den vergangenen Jahren verschiedene Lösungsansätze sowie einzelne experimentelle Prototypen entwickelt, so beispielsweise für

- die Unterstützung bei der Organisation eines geforderten Raumprogramms innerhalb einer gegebenen Grundrissstruktur [Lömker, 2006],
- die Untersuchung von Farbgestaltungsvarianten am bestehenden Bauwerk [Tonn u. Donath, 2006; Tonn, 2007] oder
- die Integration von planungsbegleitender Bauaufnahme und Entwurfsbearbeitung in CAAD-Werkzeuge [Braunes u. Donath, 2006].

Werkzeuge zur Entwurfsunterstützung können nach [Steinmann, 1997] vier Kategorien zugeordnet werden, wie Abbildung 2.5 (untere Zeile) zeigt.

Arten entwurfsunterstützender Werkzeuge

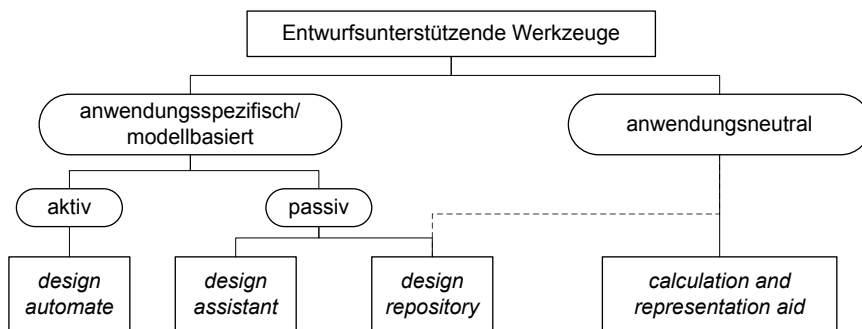


Abbildung 2.5.: Klassifikation entwurfsunterstützender Werkzeuge nach [Steinmann, 1997, S. 16]

Calculation and representation aids sind Hilfsmittel, die nicht über fachspezifische¹ Modelle verfügen und beispielsweise der Erstellung von Entwurfsdokumentationen oder der Ausführung allgemeiner Berechnungen dienen. Hierzu zählen einfache Textverarbeitung, Tabellenkalkulation oder Systeme zur Zeichnungserstellung.

Design repositories dienen vor allem dem systematischen Erfassen und Verwalten von relevanten Projektinformationen bzw. Bauwerksbeschreibungen. Sie verwenden kein fixiertes Modell, sondern lassen nutzerspezifische Erweiterungen zu.

¹ d. h. bauwesensspezifische

Design assistants verfügen über ein Modell für einen spezifischen Aufgaben- oder Anwendungsbereich. Nutzer können konkrete Ausprägungen dieses Modells erstellen, wobei die Systeme Funktionalität zur Plausibilitätsprüfung bzw. Konsistenzsicherung oder Evaluierung von Modellausprägungen zur Verfügung stellen. Dieser Gruppe sind u. a. aktuell verfügbare, mit architektonischen Bauteilen arbeitende CAAD-Systeme zuzuordnen.

Modellgeneratoren integrieren zusätzlich zum Modell Wissen zum Anwendungsbereich, das zur Konsistenzsicherung oder Vorgehensplanung eingesetzt werden kann. Modellgeneratoren als eine Untergruppe der *design automates* erlauben das Erzeugen von Entwurfs- bzw. Lösungsvorschlägen für ausgewählte formalisierbare oder durch vorab festgelegte Regeln hinsichtlich des Lösungsraumes eingeschränkte Problemstellungen. Bekannte Realisierungen beschränken sich auf experimentelle Prototypen [Steinmann, 1997]. *Modellvariatoeren* als eine zweite Untergruppe der *design automates* können durch die Veränderung von Parametern Varianten von Entwurfslösungen erzeugen. Voraussetzung dafür ist, dass die Lösung vollständig durch Regeln bestimmt werden kann. Ein typischer Anwendungsbereich solcher Werkzeuge ist die Lösung von Layoutproblemen.

Automatisierbarkeit ist nur für Spezialfälle gegeben Grundsätzlich muss davon ausgegangen werden, dass das Lösen von Entwurfsaufgaben durch Maschinen bzw. Computer nur dann möglich ist, wenn die jeweilige Aufgabe zuvor durch einen Menschen erfasst und beschrieben wurde [Krech u. a., 1992; Urchs, 2002]. Aufgrund der Vielfalt und insbesondere der Einmaligkeit von Bauprojekten und Entwurfsaufgaben kann von einer allgemeingültigen Realisierbarkeit bzw. Anwendbarkeit von Werkzeugen zum *Generieren* von Planungskonzepten nicht ausgegangen werden. Diese Aussage kann prinzipiell auf die Realisierung von Werkzeugen zur Informationsbereitstellung für Entwurfsprozesse übertragen werden, da auch diese der Einmaligkeit und Vielfalt möglicher Entwurfssituationen gerecht werden müssen.

2.6. RANDBEDINGUNGEN UND KONSEQUENZEN

Für die Planung im Kontext bestehender Bausubstanz können zusammenfassend folgende Randbedingungen festgehalten werden:

- Die frühen Planungs- bzw. Entwurfsphasen haben maßgeblichen Einfluss auf die Nachhaltigkeit eines Projektes. Dabei sind jedoch gerade diese frühen Phasen aufgrund der hohen Anteile kreativer Tätigkeiten beim Finden des Planungskonzeptes nicht vorhersehbar.
- Die Entwicklung eines dem Bestand angemessenen Planungskonzeptes erfordert die genaue Kenntnis der bestehenden Bausubstanz bzw. verlässliche Bestandsunterlagen. Aufgrund von Brüchen im Informationsfluss über die Lebensphasen eines Bauwerks hinweg werden Bestandsunterlagen aktuell durch die Bauaufnahme zur Verfügung gestellt. Der dafür notwendige Aufwand kann durch die Bereitstellung einer lebenszyklusbegleitenden digitalen Bauwerksakte stark reduziert werden.
- Das Finden des Planungskonzeptes obliegt Einzelnen, meist Architekten. Diese sind für eine disziplinübergreifende Betrachtungsweise verantwortlich und benötigen dafür disziplinübergreifenden Zugang zu bauwerksbezogenen Informationen.
- Eine allgemeingültig einsetzbare automatisierte Unterstützung von Entwurfs- und Planungsprozessen im Kontext bestehender Substanz ist aufgrund der potentiellen Vielfalt und Einmaligkeit von Entwurfs-situationen nicht möglich. Dies gilt auch für eine automatisierte, planungsbegleitende Bereitstellung von Informationen für Entwurfsprozesse.

2. PLANUNG IM KONTEXT BESTEHENDER BAUSUBSTANZ

3. STAND DER BAUWERKSMODELLIERUNG

Im vorangegangenen Kapitel wurde festgehalten, dass einmal in digitaler Form erstellte oder erfasste Informationen zu einem Bauwerk über dessen gesamten Lebenszyklus hinweg von Bedeutung sind und daher als digitale Bauwerksdokumentation oder *virtuelles Bauwerk* parallel zum real existierenden Objekt vorgehalten werden sollten. Vor diesem Hintergrund untersucht dieses Kapitel Methoden und Ansätze zur Verwaltung bauwerksbezogener Informationen, denn diese bilden die technische Grundlage für die Datenverwaltung einer digitalen Bauwerksdokumentation.

Ausgangspunkt

Inhalt

3.1. ABGRENZUNG

Produktmodelle sind strukturierte digitale Repräsentationen von Informationen zu künstlich hergestellten Objekten (Produkten). Hauptzweck ist dabei die computergestützte Verarbeitung dieser Informationen während des gesamten Lebenszyklus des repräsentierten Objekts.

*Strukturierte
Repräsentation*

Bemühungen um rechnerinterne Repräsentationen von Bauwerken im Rahmen komplexer Planungsprozesse reichen bis in die 70er Jahre zurück. Diese frühen Modelle waren zumeist implizit, d. h. in die Anwendungen integriert [Junge, 2008]. Es gab jedoch auch Bemühungen um datenbankorientierte Ansätze [Hübler, 1974]. Ab etwa den 80er Jahren gewannen vorrangig geometrieorientierte Repräsentationen sowie deren Erstellung und Bearbeitung mit Hilfe von CAD-Systemen an Bedeutung [Steinmann, 1997; East-

Historische Wurzeln

man, 1999; Borrmann, 2007]. Der Einsatz integrierter, datenbankgestützt verwalteter Modelle wurde im Zusammenhang mit CIM¹-Anwendungen insbesondere in den technischen Industriezweigen wie der Automobilindustrie, dem Schiffbau sowie dem Maschinenbau stark vorangetrieben [Willenbacher, 2002]. In den 90er Jahren fokussierten verschiedene Großprojekte auf die Übertragung dieser Ansätze auf das Bauwesen und in diesem Zusammenhang auf die Entwicklung objektorientierter Bauwerksmodelle [Björk, 1994; Hartmann, 2000; Willenbacher, 2002; Hauschild, 2003].

Eingrenzung auf objektorientierte Bauwerksmodelle Auf eine vollständige Darstellung der Entwicklungsgeschichte rechnerinterner Repräsentationen von Bauwerken wird an dieser Stelle verzichtet. Eine solche kann beispielsweise [Eastman, 1999] entnommen werden. Die Bezeichnung *Produktmodell* bzw. *Bauwerksmodell* wird im Folgenden ausschließlich für explizite, integrierte, objektorientierte Modelle verwendet. Als Grundlage der Datenverwaltung ist ausschließlich diese Form der Modelle für die Arbeit von Interesse. Die derartigen Modellen zugrunde liegenden Konzepte werden in einem separaten Abschnitt (3.3) erläutert.

3.2. BEGRIFFE

Die folgenden Abschnitte gehen auf die Definition des Modellbegriffs für den Rahmen dieser Arbeit sowie die verwendete grafische Notationsform zur Beschreibung objektorientierter Modelle ein.

3.2.1. MODELLE

Die vorliegende Arbeit tangiert verschiedene Disziplinen, in deren Fachsprache der Begriff des Modells implizit für jeweils einzelne spezifische Arten von

¹ Computer Integrated Manufacturing

Modellen verwendet wird. Aus diesem Grund ist es sinnvoll, an dieser Stelle eine explizite Definition des Begriffs vorzunehmen.

Stachowiak nennt drei wesentliche Merkmale von Modellen, die eine allgemeine, fachübergreifende Gültigkeit besitzen [Stachowiak, 1973]:

Merkmale von Modellen

- **Abbildungsmerkmal**

Modelle sind Abbildungen natürlicher oder künstlicher Originale. Die Originale können dabei selbst wiederum Modelle sein. Sie können sowohl physische Objekte als auch Bestandteil der Gedankenwelt einer Person sein.

- **Pragmatisches Merkmal**

Modelle sind jeweils für einen konkreten Verwendungszweck bestimmt. Sie werden von handelnden, modellbezogenen Subjekten für die Dauer bestimmter Zeitintervalle verwendet, um darauf bestimmte gedankliche oder tatsächliche Operationen auszuführen.

- **Verkürzungsmerkmal**

Modelle enthalten nicht alle Eigenschaften des Originals, sondern jeweils nur eine bestimmte Menge von Eigenschaften, die für den vorbestimmten Verwendungszweck und die das Modell verwendenden Subjekte relevant sind.

Aufgrund des Allgemeingültigkeitsanspruchs dieser drei Charakteristika von Modellen wird für den Rahmen dieser Arbeit die folgende Definition des Modellbegriffs zugrunde gelegt:

Def. Modell: Modelle sind abstrahierende Abbildungen natürlicher oder künstlicher Originale, die ausschließlich die für einen bestimmten Verwendungszweck relevanten Eigenschaften des Originals enthalten. *Modell*

MENTALE MODELLE UND EXTERNALISIERTE REPRÄSENTATIONEN

Modelle, die lediglich in der Gedankenwelt einer Person existieren, werden als *mentale Modelle* bezeichnet. Unter Verwendung eines bestimmten Mediums und einer bestimmten Notation können Modelle in der physischen Welt repräsentiert werden. Mit Hilfe solcher *externalisierten Repräsentationen* ist die Kommunikation der Modelle zwischen verschiedenen Personen möglich.

MODELLE IM BAUWESEN

*Unterschiedliche
Arten von Modellen*

Im Bauwesen wird eine Vielzahl von Modellen für unterschiedliche Sachverhalte bzw. Aspekte eines Bauwerks verwendet. Aus der Sicht der oben angegebenen Definition sind alle folgenden Repräsentationen von Bauwerken oder Teilen davon Modelle:

- das gedankliche Konzept eines Architekten zur Lösung einer Entwurfsaufgabe,
- zweidimensionale Planzeichnungen zur Einreichung bei der Genehmigungsbehörde oder als Grundlage für die Bauausführung,
- digitale Repräsentationen der Bauwerksgeometrie zur computergestützten Erstellung fotorealistischer Visualisierungen,
- Leistungsverzeichnisse als systematische Aufstellung aller auszuführenden Bauleistungen zur Herstellung eines Bauwerks,
- Mathematische Formeln bzw. Berechnungsalgorithmen zur Vorhersage des Verhaltens von Tragwerkselementen unter Lasteinwirkung sowie
- digital repräsentierte strukturierte Informationen für alle Lebensphasen des Bauwerks in Form eines Produkt- oder Bauwerksmodells.

In der jeweiligen Fachsprache werden alle genannten Arten von Modellen oft nur als „Modell“ bezeichnet. Im Rahmen einer fachübergreifenden Diskussion ist jedoch die implizite Voraussetzung, dass alle Beteiligten unter dem Begriff Modell auf dieselbe Art von Modellen Bezug nehmen, nicht zulässig.

3.2.2. DIE UNIFIED MODELLING LANGUAGE - UML

Die Unified Modelling Language (UML) ist eine grafische Notation (Sprache) zur Beschreibung objektorientierter Systeme. Sie definiert verschiedene Arten von Diagrammen

Etablierte grafische Notation

- zur Beschreibung der Komponenten und Strukturen eines objektorientierten Systems und
- zur Beschreibung von Verhalten und Kommunikation der Komponenten.

Gegenwärtig stellt die UML einen etablierten Standard in der objektorientierten Softwareentwicklung dar. Die ersten Versionen der Sprache entstanden in den 90er Jahren unter Federführung der drei „Väter“ Grady Booch, Ivar Jacobson und James Rumbaugh. Dabei entstand eine strukturierte Zusammenführung mehrerer bis dahin unabhängiger Notationssysteme. Von den ursprünglichen Autoren der UML finden sich detaillierte Einführungen zur Notation und Verwendung der Sprache in [Rumbaugh u. a., 1993; Booch, 1995].

Die Weiterentwicklung, Pflege und Standardisierung der UML liegt seit Ende der 90er Jahre in den Händen der Object Management Group² (OMG). In den letzten Jahren wurde die erste Version der UML grundlegend überarbeitet und als UML 2 von der OMG als neue Version des Standards

Entwicklungsstand

² siehe <http://www.omg.org>

freigegeben. Derzeit aktuell ist die Version 2.1.2 vom November 2007.³ Zur Einführung in die Notation der UML 2 siehe z.B. [Störrle, 2005]. Zur objektorientierten Modellierung siehe darüber hinaus auch [Balzert, 2005].

Verwendung in der Arbeit

Die UML dient insbesondere ab Stufe III der Arbeit als grafische Notation zur Beschreibung objektorientierter Datenstrukturen und Systemkomponenten. Dabei kommen einerseits *Klassen- und Objektdiagramme*⁴ zur Beschreibung von Systemstrukturen zum Einsatz. Andererseits werden *Sequenzdiagramme*⁵ zur Beschreibung der Interaktion mehrerer Systemkomponenten verwendet. Ferner dienen *Aktivitätsdiagramme*⁶ der Darstellung von Arbeitsabläufen.

3.3. PRODUKTMODELLE IM BAUWESEN - BAUWERKSMODELLE

Integrierte Produktmodelle

Nach [Polly, 1996] enthält ein *integriertes Produktmodell* alle relevanten Produktmerkmale, die in den einzelnen Lebensphasen des Produktes entstehen. Die Basis dafür bildet eine einheitliche, allgemeine, lebensphasenübergreifende und redundanzfreie Grundstruktur. Diese wird als *konzeptuelles Schema* bezeichnet und ist explizit beschrieben. Die Informationsverwaltung erfolgt dabei in der Regel *datenbankgestützt*, einzelne Dokumente werden als Sichten auf die Datenbasis generiert. Dieser Ansatz steht im Gegensatz zur traditionellen Art der Repräsentation von Produkten in Form einzelner, unabhängiger Dokumente und deren Bearbeitung mit Hilfe von Applikationen, die ausschließlich implizite Modelle verwenden.

Digitale Bauwerksmodelle

Produktmodelle im Bauwesen, auch als digitale Bauwerksmodelle be-

³ Zu den aktuellen Spezifikationen siehe <http://www.omg.org/spec/UML/2.1.2/>

⁴ [Booch, 1995, S. 223 ff.] sowie zur UML 2 [Störrle, 2005, S. 43 ff.]

⁵ [Booch, 1995, S. 272 ff.] bzw. zur UML 2 [Störrle, 2005, S. 222 ff.]

⁶ [Störrle, 2005, S. 193 ff.]

zeichnet, basieren nach [Willenbacher, 2002] auf einem *objektorientiert modellierten konzeptuellen Schema*, welches die lebenszyklusweit erforderlichen Daten in ihrer Gesamtheit strukturiert beschreibt. Zur Definition bzw. Verwendung des Produktmodellbegriffs im Bauwesen vgl. auch [Björk, 1995; Rüppel, 2007]. Eine Systematisierung der Anforderungen an digitale Bauwerksmodelle wurde in [Björk, 1995] vorgenommen. In [Eastman, 1999] findet sich ein Überblick über die Entwicklung der Bauwerksmodellierung bis zum Ende der 90er Jahre.

3.3.1. MODELLIERUNGSPARADIGMA

Als theoretische Wurzeln der Produktmodellierung können einerseits die konzeptuelle Modellierung von Datenbankanwendungen und andererseits die Objektorientierung gesehen werden. Beide Ansätze stützen sich wiederum auf Konzepte der formalen Repräsentation von Wissen. Dazu vgl. [Björk, 1995; Eastman, 1999]. Zur konzeptuellen Modellierung siehe [Elmasri u. Navathe, 2002; Leser u. Naumann, 2007], zur Objektorientierung vgl. [Rumbaugh u. a., 1993; Booch, 1995; Balzert, 2005].

Theoretische Wurzeln

Der Ansatz der konzeptuellen Modellierung stammt aus der Datenbanktechnik, die Objektorientierung ist ein Konzept der Softwareentwicklung. Beide Ansätze stimmen in einem wesentlichen Grundprinzip überein, das gleichermaßen auch für die Produktmodellierung gilt. Es wird dabei strikt zwischen den beobachtbaren Phänomenen innerhalb des betrachteten Anwendungsbereichs und abstrakten Konzepten zur Beschreibung dieser Phänomene unterschieden. In der Objektorientierung werden die Phänomene als *Objekte* oder *Exemplare* bezeichnet, abstrakte Konzepte als *Klassen*.

Unterscheidung von Phänomenen und Konzepten

Informationen, die einzelne beobachtete Phänomene bzw. Objekte beschreiben, fasst Björk unter dem englischen Begriff *information base* zusammen [Björk, 1995]. Dies sind die Informationen, die später von einer

Phänomene bzw. Objekte

Softwareapplikation bearbeitet oder einer Datenbank verwaltet werden sollen (z. B. „Wand-005 ist 4,50 m lang.“).

*Konzepte bzw.
Klassen als Struktur
zur Verwaltung von
Objekten*

Zur Abbildung und Verwaltung dieser Informationen sind entsprechende Datenstrukturen erforderlich, die die für den Anwendungsbereich gültigen Konzepte und deren Eigenschaften definieren (z. B. „Wände haben eine Dimension ‚Länge‘.“). Eine solche Datenstruktur besteht in der Regel aus einer Menge gültiger Typen, die die Konzepte des Anwendungsbereiches repräsentieren, einer Menge von Attributen, die die relevanten Eigenschaften der Konzepte beschreiben sowie der Spezifikation von Beziehungen, die zwischen einzelnen Exemplaren der Konzepte bestehen können. Zusätzlich ist die Festlegung von Regeln für die Ausprägung der Struktur und die Konsistenz der Informationen möglich. Für diese Datenstruktur verwendet Björk die englische Bezeichnung *conceptual schema*. Im Deutschen werden synonym die Begriffe *konzeptuelles Schema* bzw. *konzeptionelle Sicht* verwendet [Leser u. Naumann, 2007]. Eine einheitlich verwendete deutsche Bezeichnung zu *information base* existiert nicht.

3.3.2. MODELLSCHEMATA UND MODELLEDATEN

*Zwei
Modellierungsebenen
in Produktmodellen*

Ausgehend von der auf Seite 37 angegebenen Definition des Modellbegriffs muss in Bezug auf die Produktmodellierung zwischen zwei Modellierungsebenen unterschieden werden:

- Die Festlegung der verwendeten Konzepte und der für sie gültigen Eigenschaften in Form eines konzeptuellen Schemas stellt ein *Modell des Anwendungsbereichs* dar.
- Informationen zu beobachteten Phänomenen als Ausprägung des konzeptuellen Schemas stellen *Modelle einzelner konkreter Objekte* dar.

Der Begriff des Produktmodells findet diesbezüglich in der Literatur keine durchgehend eindeutige Verwendung. Einige Autoren verstehen unter Produkt- bzw. Bauwerksmodellen konzeptuelle Schemata, also Modelle des Anwendungsbereichs [Rüppel, 2007; Willenbacher, 2002]. Bei anderen Autoren steht der Begriff für einzelne Ausprägungen konzeptueller Schemata und damit für Modelle konkreter Objekte [Hannus u. a., 1995; Björk, 1995]. Manche Autoren grenzen für die jeweils andere Modellierungsebene explizit einen zweiten Begriff ab. Bezogen auf Produktmodelle im Bauwesen schlagen Björk und Willenbacher die in Tabelle 3.1 gezeigte Verwendung der Bezeichnungen vor.

Zuordnung von Begriffen

	[Björk, 1995]	[Willenbacher, 2002]
<i>conceptual schema</i>	building product data model	Bauwerksmodell
<i>information base</i>	building product model	Bauwerksmodelldaten

Tabelle 3.1.: Bezeichnungen der unterschiedlichen Modellierungsebenen

Björk empfiehlt die Verwendung von *product model* auch zur Bezeichnung des Produktmodellansatzes in seiner Gesamtheit. In Anlehnung an diese Empfehlung sowie Bezug nehmend auf die verwendete Definition des Begriffs Modell werden im Rahmen dieser Arbeit die Bezeichnungen so verwendet wie in Tabelle 3.2 dargestellt, wobei ein Modell ein Produkt-, Bauwerks- oder Domänenmodell sein kann. Diese explizite Unterscheidung ist notwendig, da der in der Arbeit entwickelte Ansatz nicht ausschließlich auf eine der beiden genannten Modellierungsebenen fokussiert.

Schema und Daten

<i>conceptual schema</i>	<i>information base</i>
Modellschema	Modelldaten
Modell	

Tabelle 3.2.: Verwendete Bezeichnungen für die Modellierungsebenen

3.4. DER PRODUKTMODELLANSATZ ALS METHODE DER INTEGRATION

*Stark heterogene
Softwarelandschaft*

Die traditionell gewachsene Softwarelandschaft im Bauwesen ist in hohem Maße durch Heterogenität und Inkompatibilität der einzelnen Anwendungen gekennzeichnet. Die verfügbaren Anwendungen betrachten dabei jeweils einzelne Aufgaben oder Teilbereiche der Planung herausgelöst aus dem Gesamtprozess und verwalten die dafür relevanten Informationen in nativen Datenstrukturen.⁷ Zu den kritischen Merkmalen, die die heutige Praxis zum Teil nach wie vor prägen, zählen unter anderem die folgenden Punkte:

- Eine Vielzahl heterogener Hard- und Softwareplattformen wird verwendet.
- Die einzelnen Anwendungen speichern ihre Daten in eigenen, nativen Datenformaten.
- Die verfügbaren Applikationen unterstützen ausschließlich Teilaufgaben in Form von Insellösungen.
- Die Applikationen übertragen lediglich die traditionellen Arbeitsweisen zur Erstellung lose zusammenhängender Einzeldokumente auf das Medium Computer.

⁷ M. Hannus illustrierte diese Situation anschaulich als *islands of automation in construction*. Siehe <http://cic.vtt.fi/hannus/islands/index.html>

- Der Austausch von Informationen zwischen einzelnen fachlich Beteiligten bzw. deren Fachapplikationen ist nicht ohne manuelles Eingreifen bei der Interpretation und Eingabe der Informationen möglich.

Unter diesen Voraussetzungen ist ein durchgängiger Informationsfluss zwischen den verwendeten Fachapplikationen über alle Phasen der Bauplanung oder gar über alle Lebensphasen eines Bauwerkes hinweg nicht realisierbar. Vielmehr stellt der Austausch von Planungsinformationen zum Zweck der Weiterbearbeitung durch unterschiedliche Fachapplikationen einen hochgradig ineffizienten und fehleranfälligen Aspekt des gesamten Planungsprozesses dar. Eastman schreibt in diesem Zusammenhang:

*Kein durchgängiger
Informationsfluss*

„Integration requires access and incorporation of appropriate data, interpretation of results and possibly iterative reuse and exchange with other members of the building team.” [Eastman, 1999, S. 6]

Eine derartige Integration in Form eines automatisierten, durchgängig computergestützt abgewickelten Informationsaustauschs zwischen verschiedenen Fachapplikationen ist theoretisch auf der Basis eines integrierenden, lebensphasenübergreifenden Bauwerksmodellschemas nach dem Ansatz integrierter Produktmodelle möglich. Dazu formuliert Eastman folgendermaßen:

*Lösungsansatz
integriertes
Produktmodell*

„The immediate challenge is to develop a digital representation of a building project that can first be used for feasibility, then design, then fabrication, and then operations and maintenance, and will enable all parties to do their work more effectively than now. The representation must facilitate use of computer-based tools and digital communication.” [Eastman, 1999, S. 31]

„The challenge before us is to develop an electronic representation of a building in a form capable of supporting all major activities throughout the building life cycle.” [Eastman, 1999, S. 72]

3.5. MERKMALE UND ANSÄTZE INTEGRIERTER INFORMATIONSSYSTEME

Willenbacher [Willenbacher, 2002] unterscheidet Bezug nehmend auf [Bakkeren u. Tolman, 1995] drei Richtungen der Integration im Bauwesen (vgl. Abbildung 3.1):

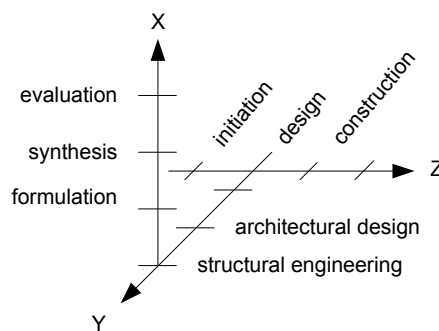


Abbildung 3.1.: Integrationsrichtungen nach [Bakkeren u. Tolman, 1995]

- X (intradisziplinär): Integration der Arbeitsschritte einer Disziplin bzw. Tätigkeit
- Y (interdisziplinär): Integration der Tätigkeiten verschiedener Disziplinen
- Z (phasenübergreifend): Integration verschiedener Phasen des Bauwerkslebenszyklus.

Wachsende Komplexität Grundsätzlich nimmt mit jeder einzubeziehenden Integrationsrichtung der Umfang der potentiell abzubildenden Informationen und Abhängigkeiten zu. Somit wächst auch die Komplexität der zu entwickelnden Integrationslösung.

Zur Entwicklung integrierter Informationssysteme existiert derzeit ein breites Spektrum verschiedener Ansätze, Vorgehensweisen und Systemarchitekturen. Eine umfassende Darstellung dieses Themenkomplexes soll nicht Gegenstand dieser Arbeit sein. An dieser Stelle werden lediglich einige grundsätzliche Aspekte eingeführt, die zum Verständnis und zur Bewertung der in den folgenden Abschnitten diskutierten Integrationsansätze im Bauwesen von Bedeutung sind. Interessierte Leser seien zur vertiefenden Lektüre an [Leser u. Naumann, 2007] verwiesen. In der genannten Quelle werden eine Reihe von Merkmalen integrierter Informationssysteme erläutert, die ausschlaggebend für die Bewertung und Auswahl eines konkreten Lösungsansatzes für ein gegebenes Problem sind. Zur Charakterisierung der im Folgenden besprochenen Integrationsansätze sind hauptsächlich drei Merkmale relevant:

*Relevante Merkmale
integrierter
Informationssysteme*

- die angestrebte bzw. realisierte Kopplung der zu integrierenden Datenquellen
- der Ansatzpunkt und die Vorgehensweise beim Systementwurf (Entwurfsstrategie)
- der Ort der Datenspeicherung.

3.5.1. ART DER KOPPLUNG

Die Kopplung von Datenquellen kann durch ein globales integrierendes Schema realisiert werden, anhand dessen das System Anfragen entgegennimmt. Die lokalen Schemata der Datenquellen werden jeweils durch *Mappings* mit dem globalen Schema in Beziehung gesetzt. Unter Mappings wird in diesem Zusammenhang eine Menge explizit beschriebener Korrespondenzen zwischen den Konzepten und/ oder Attributen zweier unterschiedlicher Schemata verstanden. Mappings bilden unter anderem die Voraussetzung für automatisierte Datentransformationen zwischen zwei unterschiedlichen

*enge Kopplung über
globales Schema*

Schemata. Mit Hilfe des globalen Schemas und der Mappings kann das System selbst Heterogenität zwischen den integrierten Datenquellen überbrücken und die Datenquellen nach außen als integrierten Gesamtdatenbestand zur Verfügung stellen. Diese Art der Kopplung, basierend auf einem globalen Schema, wird als *enge Kopplung* bezeichnet.

lose Kopplung *Lose gekoppelte* Systeme arbeiten dagegen ohne globales integrierendes Schema. Eine Kopplung wird dann beispielsweise durch eine strukturierte Anfragesprache erreicht, mit der mehrere Schemata gleichzeitig in einer Anfrage angesprochen werden können. Die Überwindung der Heterogenität bleibt dann jedoch Aufgabe des Informationssuchenden, denn diesem allein bleibt es überlassen, entsprechend komplexe Anfragen korrekt zu formulieren.

Peer-DBMS Eine andere Art der losen Kopplung verwenden Peer-Datenbankmanagementsysteme (Peer-DBMS). Bei diesen Systemen stellen einzelne Peers eine oder mehrere Datenquellen anhand eines lokalen Schemas für andere Peers zur Verfügung. Das eigene Schema wird dabei durch Mappings mit den Schemata der Nachbar-Peers in Beziehung gesetzt. Peers können anhand ihres Schemas Anfragen bearbeiten, diese gegebenenfalls an Nachbarn weiterleiten, sowie Daten für von Nachbarn bearbeitete Anfragen zur Verfügung stellen. Ein globales, alle Peers integrierendes Schema existiert in diesem Fall nicht. Jeder Peer kennt nur seine unmittelbaren Nachbarn, zu deren Schemata Mappings definiert wurden.

3.5.2. ENTWURFSSTRATEGIE

Bottom-up Wird die vollständige Integration einer bekannten Menge von Datenquellen angestrebt, wobei die Struktur und Anzahl der lokalen Schemata nicht variiert, ist eine *Bottom-Up*-Entwicklung sinnvoll. Ausgehend von den lokalen Schemata der Quellen wird ein globales integrierendes Schema entwickelt,

das alle Konzepte der lokalen Schemata redundanzfrei enthält. Dieser Arbeitsschritt wird als Schemaintegration bezeichnet. Über das integrierende Schema kann später auf die Daten der einzelnen Quellen zugegriffen werden.

Die umgekehrte Vorgehensweise ist sinnvoll, wenn im Voraus die Anzahl und Struktur der zu integrierenden lokalen Schemata nicht bekannt und/oder mit einem häufigen Wechsel zu rechnen ist. Dann wird anhand der potentiell bereitzustellenden Informationen zunächst das globale integrierende Schema entworfen. Die relevanten Quellen werden erst danach ausgewählt. Deren Schemata müssen dann durch Mappings mit dem zuvor entwickelten globalen Schema in Beziehung gesetzt werden. In diesem Fall muss jedoch in Kauf genommen werden, dass sich nicht alle lokalen Schemata potentiell zu integrierender Quellen vollständig auf das globale Schema abbilden lassen. Diese zweite Vorgehensweise wird als *Top-Down-Entwurf* bezeichnet. *Top-down*

3.5.3. ORT DER DATENSPEICHERUNG

Werden die Daten der einzelnen Quellen transformiert und persistent im globalen Schema gespeichert, wird die Form der Integration als *materialisiert* bezeichnet. *Materialisierte Integration*

Die Daten können jedoch auch weiterhin ausschließlich durch die Quellen persistent verwaltet werden. Transformationen in das globale Schema erfolgen dann nur temporär zum Zeitpunkt der Anfragebearbeitung. In diesem Fall ist die Integration nur *virtuell* realisiert. *Virtuelle Integration*

3.6. INTEGRATIONSANSÄTZE IM BAUWESEN

Für die Integration mit enger Kopplung ist in jedem Fall ein globales integrierendes Schema zu entwickeln. Nach dem Bottom-Up-Verfahren ist dies *Wahl der Entwurfsstrategie*

jedoch nur für Ausschnitte des gesamten Anwendungsbereiches realisierbar – beispielsweise entlang einer einzelnen der drei auf Seite 46 genannten Integrationsrichtungen. Ein wesentliches Problem bei diesem Verfahren stellt die Vielzahl existierender Bauwerkstypen, verfügbarer Technologien und gültiger Normen sowie die Vielfalt und Heterogenität der existierenden Softwarelandschaft dar. Für eine Integration über fachliche, projekt- und lebensphasenbezogene Grenzen hinweg kommt somit nur eine Top-Down-Strategie in Betracht.

Entwicklung eines geschlossenen, allgemein gültigen Schemas ist problematisch Die Entwicklung eines global integrierenden Schemas bildete einen wesentlichen Inhalt der frühen Forschungsarbeiten zu Produktmodellen für das Bauwesen. Ende der 90er Jahre setzte sich im Ergebnis verschiedener Großprojekte die Auffassung durch, dass die Entwicklung eines vollständigen, allgemein gültigen standardisierten Schemas⁸ eine unrealistische Aufgabe sei [Hannus u. a., 1995; Björk, 1995; Eastman, 1999].

Zerlegung in Teilschemata Anstelle eines einzelnen allumfassenden Schemas wurde von verschiedenen Autoren eine Zerlegung in mehrere Teilschemata und deren hierarchische Organisation auf mehreren Ebenen vorgeschlagen (siehe Tabelle 3.3). Das Ebenenkonzept fand im Wesentlichen Eingang in die Spezifikation des Industry Foundation Classes (IFC)⁹, die mittlerweile als standardisiertes, fachübergreifendes Bauwerksmodellschema zur Verfügung stehen [IAI, 2004, 2000].

Die folgenden Erläuterungen der einzelnen Ebenen beziehen sich im Wesentlichen auf [Björk, 1995; IAI, 2000, 2004]. Die in [Hannus u. a., 1995] beschriebenen Ebenen stimmen inhaltlich prinzipiell mit denen der IFC überein – mit Ausnahme der bei Hannus u. a. fehlenden separaten Ressour-

⁸ d. h. eines Schemas, das die benötigten Informationen für alle Arten von Bauwerken, alle fachspezifischen Aspekte und alle Bauwerkslebensphasen abbildet

⁹ Die aktuellen Spezifikationen finden sich unter [http://www.iai-international.org/Model/IFC\(ifcXML\)Specs.html](http://www.iai-international.org/Model/IFC(ifcXML)Specs.html).

<i>Ebene</i>	[Hannus u. a., 1995]	u. a.,	[Björk, 1995]	Industry Foundation Classes (IFC)	[Willenbacher, 2002]
0	common model	meta	information modelling language	EXPRESS/-XML	Metamodell
1	-		generic product description model	resources layer	-
2	common model	core	building kernel model	kernel + kernel extentions = core layer; interoperability layer	-
3	application domain models	do-	aspect models	domain layer	Domänenmodelle
4	-		appliacion models	-	-

Tabelle 3.3.: Bezeichnungen der Ebenen in verschiedenen Modellierungsansätzen

cenebene. Die gemeinsame Darstellung mit dem in [Willenbacher, 2002; Hauschild, 2003] verfolgten Ansatz dient primär der Einordnung aller hier zitierten Ansätze in ein Gesamtbild. Auf spezifische Besonderheiten und Unterschiede wird an entsprechender Stelle in den folgenden Abschnitten eingegangen.

*Modellierungs-
konzepte* **Ebene 0** dient bei allen Ansätzen der Festlegung der verwendeten Modellierungskonzepte. In der Regel erfolgt dies implizit durch die Festlegung des verwendeten Modellierungsparadigmas und/oder die Festlegung der verwendeten Notation bzw. Modellierungssprache.

Ressourcen **Ebene 1** legt gemeinsam verwendete Ressourcen fest, die unabhängig von speziellen Konzepten zur Beschreibung der Bauwerksstruktur oder zur Beschreibung fachspezifischer Informationen dienen. Dies sind beispielsweise Typen zur Beschreibung von Geometrie, Materialien oder Maßeinheiten. Die Schemata und Konzepte der Ebene 1 können von allen höheren Ebenen durch Referenzieren derselben verwendet werden.

*Gemeinsam
verwendete Konzepte* **Ebene 2** definiert solche Konzepte zur Beschreibung von Bauwerken bzw. Bauprojekten, die von allen Fachapplikationen gemeinsam verwendet werden, bzw. Konzepten, über die im Verlauf des Bauwerkslebenszyklus Informationen auszutauschen sind. Diese Konzepte können von denen höherer Ebenen übernommen und durch Spezialisierung erweitert werden.

*Fachspezifische
Konzepte* **Ebene 3** enthält Konzepte, die ausschließlich für die Fachapplikationen einzelner Disziplinen und/oder Lebensphasen des Bauwerks relevant sind. Die Konzepte sind dabei in domänen- bzw. lebensphasenspezifischen Schemata organisiert. Die Anzahl dieser spezifischen Schemata ist prinzipiell nicht begrenzt. In allen Ansätzen, in denen die Ebene 2 besetzt ist, ist die Entwicklung der Schemata der Ebene 3 durch eine Spezialisierung von Konzepten der Ebene 2 vorgesehen. Die spezifischen Schemata können dabei die Menge der von Ebene 2 übernommenen Konzepte individuell festlegen.

Darüber hinaus können eigene fachspezifische Konzepte ergänzt werden.

Ebene 4 wird nur von Björk erwähnt und ist nicht mehr Gegenstand eines fachübergreifenden integrierenden Bauwerksmodellschemas. Diese Ebene steht für alle konzeptuellen Schemata, die durch individuelle Softwareapplikationen implementiert werden, und umfasst quasi alle nativen Schemata.

*Applikations-
spezifische
Konzepte*

Der von Willenbacher und Hauschild entwickelte Ansatz verzichtet dagegen auf die Verwendung eines integrierenden globalen Schemas. Integration wird stattdessen in loser Form nach dem Muster der Peer-DBMS realisiert [Willenbacher, 2002; Hauschild, 2003].

Gegenstand der drei übrigen Ansätze ist zunächst nur der Entwurf eines globalen integrierenden Schemas. Aussagen bezüglich eines das Schema verwendenden integrierten Systems werden dabei nicht getroffen.

Bezogen auf die IFC zeichnen sich in der Praxis derzeit zwei alternative Ansätze ab. Der erste Ansatz betrachtet die IFC lediglich als standardisiertes Austauschformat für einen dateibasierten Datenaustausch. Verschiedene Softwarehersteller bieten mittlerweile entsprechende Import-/Export-schnittstellen an. Eine Übersicht der aktuell IFC zertifizierten Softwareprodukte findet sich auf den Seiten der Industrieallianz für Interoperabilität (IAI)¹⁰. Der zweite Ansatz verwendet die IFC tatsächlich als integrierendes Schema für eine gemeinsame Datenhaltung durch einen Modellserver. Derzeit bieten zwei Hersteller entsprechende kommerzielle Produkte¹¹ an.

*IFC-basierte
Integration*

Tabelle 3.4 stellt die Ansätze hinsichtlich der oben erläuterten Merkmale integrierter Informationssysteme gegenüber.

¹⁰ Siehe http://www.buildingsmart.de/2/2_01_01.htm

¹¹ EPM Technology (<http://www.epmtech.jotne.com>) und eurostep (<http://www.eurostep.com>)

3. STAND DER BAUWERKSMODELLIERUNG

	IFC-basierte Integration		Verbund von Teilmodellen
	IFC als Austauschformat für dateibasierten Austausch	IFC als Schema für gemeinsamen Modellserver	nach dem Muster von Peer-DBMS
<i>Globales integrierendes Schema / Art der Kopplung</i>	Enge Kopplung durch ein globales integrierendes Schema		Kein globales integrierendes Schema; Integration durch lokale Peer-Mappings
<i>Gemeinsamer integrierter Datenbestand / Ort der Datenspeicherung</i>	Daten verbleiben im lokalen Dateisystem der Beteiligten und sind nicht über das globale Schema zugreifbar	Gemeinsamer integrierter Datenbestand; materialisierte Integration	Daten verbleiben lokal bei den einzelnen Peers, sind aber über Mappings zugreifbar; virtuelle Integration
<i>Erforderliche Infrastruktur</i>	Nicht erforderlich	erforderlich in Form des gemeinsamen Modellservers	Erforderlich in Form der Adaption der einzelnen Peers für die Integration in den Verbund
<i>Inhaltliche Erweiterbarkeit des Gesamtdatenbestandes/ Einfügen neuer Quellen</i>	Nicht im globalen Schema abbildbare Informationen können nicht ausgetauscht bzw. gemeinsam verwendet werden		Möglich durch Einfügen neuer Peers und Definition entsprechender Mappings
<i>Zentraler Einstiegspunkt für Informationssuche</i>	Nicht vorhanden (verstreute, nicht explizit zusammenhängende Datenquellen)	Vorhanden: Modellserver	Nicht vorhanden, jedoch über Mappings explizit verknüpfte Quellen

Tabelle 3.4.: Technische Merkmale der Integrationsansätze

Dateibasierter Austausch entspricht etwa den traditionellen Arbeitsweisen zur Erstellung lose zusammenhängender Einzeldokumente, die auf Anfrage der Beteiligten weitergegeben werden. Der Ansatz stellt keine Anforderungen bezüglich zusätzlich benötigter Infrastruktur. Somit ist eine Realisierung auch in kleinen Projekten hinsichtlich des materiellen Aufwands unproblematisch, sofern entsprechende Fachapplikationen mit Import-/Exportschnittstellen zur Verfügung stehen. Die originalen Planungsdaten verbleiben dabei lokal im Dateisystem der einzelnen Planungsbeteiligten. Das Erzeugen und Einlesen der Austauschdateien erfordert in der Regel einen manuellen Eingriff. Zwar verwenden die beteiligten Fachapplikationen das globale Bauwerksmodellschema, dennoch existiert kein gemeinsamer Datenbestand, aus dem die einzelnen Arbeitsdokumente als Sichten auf die Daten erzeugt werden.

*Dateibasierter
Austausch*

Für eine Fortschreibung der Daten nach Abschluss eines Neubauprojektes über den gesamten Lebenszyklus des Bauwerks hinweg ist diese Integrationsform kaum geeignet. Einerseits kann aufgrund der verstreuten, nicht explizit zusammenhängenden Datenquellen die Verfügbarkeit und Zugreifbarkeit über den gesamten Bauwerkslebenszyklus organisatorisch nur sehr schwer sichergestellt werden. Andererseits ist keinerlei zentraler Einstiegspunkt für die Suche in einem solchen fragmentierten Datenbestand gegeben. Nach Abschluss der Planungs- und Bauphase und Auflösung des Planungsteams geht auch das Wissen darüber, welche Informationen wo gespeichert sind, verloren.

*Eignung für
langfristige
Datenhaltung*

Ein gemeinsam verwendeter Modellserver verwaltet auf Basis der IFC als globales integrierendes Schemas einen materialisierten gemeinsamen Datenbestand. Bei diesem Ansatz wird tatsächlich ein integriertes produktmodellbasiertes Informationssystem realisiert. Die Verfügbarkeit und Zugreifbarkeit des Datenbestandes – auch über lange Zeiträume hinweg – stellt prinzipiell kein organisatorisches Problem dar. Ein zentraler Einstiegspunkt

*Gemeinsam genutzter
Modellserver*

für die Informationssuche ist über den Server ebenfalls gegeben. Allerdings stellt der Modellserver eine zusätzlich benötigte Infrastrukturkomponente dar, für die zusätzliche Aufwendungen zur Anschaffung, Einrichtung, Betrieb, Pflege und Wartung erforderlich sind.

Grenzen von IFC Für beide Ansätze gilt, dass die beteiligten datenbearbeitenden Fachapplikationen das globale Schema¹² für Import und Export unterstützen müssen. Es bleibt in der Verantwortung der Fachapplikationen, entweder das globale integrierende Schema direkt zu implementieren oder Mappings zwischen dem eigenen internen Schema und einer entsprechenden Import-/ Export-schnittstelle zu definieren. Informationen, die nicht Bestandteil des globalen Schemas sind, können nicht ausgetauscht bzw. in den gemeinsamen Datenbestand übernommen werden. Problematisch ist dies für die Integration regional spezifischer Normen oder technischer Regeln in die rechnerinterne Repräsentation des Bauwerkes. Entsprechende Informationsquellen können lediglich durch Referenzen mit dem integrierten Datenbestand verknüpft werden.

Komplexität Umfassende Bauwerksmodellschemata wie die IFC erreichen aufgrund des Allgemeingültigkeitsanspruchs unweigerlich ein hohes Maß an Komplexität. Die aktuelle Version IFC 2x3 enthält mehr als 44 Teilschemata und mehr als 600 Konzepte zur Beschreibung von Entitäten sowie weitere einfache Datentypen¹³. Dementsprechend hoch ist auch der Aufwand zur Implementierung des Standards durch die Fachapplikationen. Für einzelne Personen ist es nahezu unmöglich, den Überblick über das Gesamtschema einschließlich aller Konzepte und Beziehungen zu behalten. Die Definition von Mappings zwischen applikationsspezifischen Schemata und den IFC stellt somit eine anspruchsvolle Aufgabe dar.

¹² oder zumindest die benötigten Teile davon

¹³ Zu den aktuellen Spezifikationen siehe [http://www.iai-international.org/Model/IFC\(ifcXML\)Specs.html](http://www.iai-international.org/Model/IFC(ifcXML)Specs.html)

Generell sind tiefgreifende Veränderungen wie die Einführung eines gemeinsam verwendeten integrierenden Modellschemas aufgrund der kleinteiligen organisatorischen Struktur der Bauwirtschaft, wenn überhaupt, nur langfristig möglich [Eastman, 1999]. Mit einer kurzfristigen Etablierung der IFC als exklusiv verwendetes global integrierendes Schema ist somit nicht zu rechnen.

*Langfristige
Veränderung*

Ein Verbund gleichberechtigter Teilmodelle nach dem Muster von Peer-DBMS definiert zwar kein globales integrierendes Schema, dennoch ist die Integration auf der Basis der Mappings zwischen den Peers gegeben (vgl. Abbildung 3.2). Die einzelnen Beteiligten behalten die Hoheit über ihre jeweiligen Daten, veröffentlichen diese jedoch über das lokale Schema ihres jeweiligen Peers. Somit ist die Zugreifbarkeit gegeben. Die Verfügbarkeit hängt jedoch von der Aktivität der einzelnen Peers ab, was über den langen Zeitraum des Bauwerkslebenszyklus nicht unproblematisch sein kann.

*Variabler
Modellverbund*

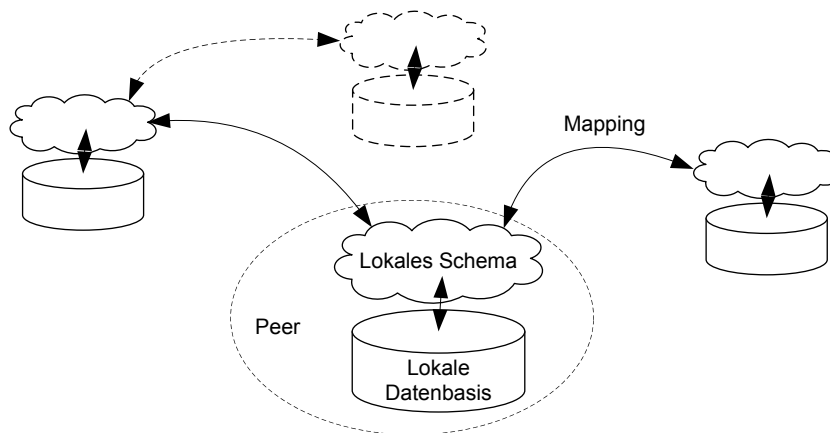


Abbildung 3.2.: Prinzip von Peer-DBMS

Der gemeinsame Datenbestand existiert bei einem Modellverbund nur virtuell. Transformationen zwischen einzelnen Peers erfolgen jeweils auf Anfrage. Sie können jedoch anhand der Mappings automatisiert erfolgen und

*Struktur und
Integration*

erfordern in der Regel kein manuelles Eingreifen wie beim dateibasierten Austausch. Darüber hinaus ist die Zugriffsgranularität wesentlich feiner. Der Aufwand für zusätzlich benötigte Infrastrukturkomponenten verteilt sich besser als beim Modellserveransatz, da jeweils nur die einzelnen Peers für die Teilnahme an der Kommunikation im Verbund adaptiert werden müssen. Dies kann durch so genannte *Wrapper* erfolgen, die eine Umhüllung für bereits bestehende Komponenten bilden und nach außen die für die Teilnahme im Verbund benötigten Schnittstellen bereitstellen. Auf diese Weise ist auch die inhaltliche Erweiterung des Verbundes und die Integration projektspezifischer Informationen möglich.

Flexibilität Diese Flexibilität und Erweiterbarkeit des Verbundes stellt einen bedeutenden Vorteil des Ansatzes dar. In [Rüppel u. a., 2006, 2007] wird ein solcher Modellverbund zur Demonstration kooperativer Gebäudeplanung beschrieben, wobei die Planung des baulichen Brandschutzes für einen Gebäudeentwurf im Mittelpunkt steht. Brandschutzregeln, die jeweils vom spezifischen Gebäudetyp abhängig sind und in Deutschland außerdem regional variieren, werden in Form eines eigenständigen Brandschutzregelmodells in den Verbund integriert. Auf diese Weise ist es leicht möglich, das Regelmodell entsprechend projektspezifischer Anforderungen auszutauschen, da lediglich lokale Anpassungen in Form neuer Mappings erforderlich sind. Die übrigen am Verbund beteiligten Peers werden dabei nicht beeinflusst.

*Grenzen eines
variablen
Modellverbundes* In Abhängigkeit von der Anzahl der beteiligten Peers und den zwischen ihnen abzubildenden Beziehungen kann die Struktur eines solchen Modellverbundes sehr komplex und damit unüberschaubar werden. Beim Einfügen neuer Peers müssen theoretisch alle lokalen Schemata der bereits vorhandenen Peers bekannt sein, um relevante Nachbar-Peers zu identifizieren und über Mappings verknüpfen zu können. Anfragen werden gegebenenfalls über mehrere Peers weitergereicht, so dass kaum oder gar nicht nachvollziehbar ist, welche Peers zur Beantwortung beigetragen haben [Leser u. Naumann,

2007]. Ein bedeutendes Problem stellt auch das Fehlen eines zentralen Einstiegspunktes für die Informationssuche dar. Für Personen, die nicht über detaillierte Kenntnisse der Gesamtstruktur sowie der einzelnen beteiligten Schemata verfügen, ist das Auffinden von Informationen somit nahezu unmöglich.

Die hier exemplarisch beschriebenen Ansätze zur Informationsintegration im Bauwesen dienen in erster Linie der Markierung des insgesamt existierenden Spektrums an Lösungsansätzen, nicht dessen umfassender Beschreibung. Derzeit zeichnet sich kein allgemeiner Konsens hinsichtlich eines besonders geeigneten oder bevorzugten Ansatzes ab. Die verschiedenen Integrationsansätze existieren vielmehr ebenso parallel wie die verschiedenen Fachapplikationen in der historisch gewachsenen heterogenen Softwarelandschaft im Bauwesen. Aufgrund der relativ jungen Entwicklung fehlen derzeit Langzeiterfahrungen zum Einsatz der beschriebenen Integrationsansätze über den gesamten Bauwerkslebenszyklus hinweg. Literatur zu produktmodellbasierter Integration fokussiert implizit meist die Neubauplanung und gegebenenfalls die eigentliche Bauphase. Die langen Zeiträume der Gebäudenutzung und eventuelle Baumaßnahmen am bestehenden Gebäude werden in der Regel vernachlässigt [GISMO Projekt, 2008].

*Kein Konsens bzgl.
Datenverwaltung*

3.7. DYNAMISCHE BAUWERKSMODELLE

Aufgrund der spezifischen Anforderungen, insbesondere vor dem Hintergrund der extrem langen Modelllebensdauer (= Lebensdauer des Bauwerks) schlagen verschiedene Autoren eine Dynamisierung der Modelle vor. Dynamik bezieht sich dabei nicht auf die Modelldaten, denn die Variabilität der Ausprägungen ist ein grundsätzliches Merkmal des objektorientierten Paradigmas. Dynamik wird stattdessen bezogen auf die Modellschemata verstanden. Das bedeutet, während der gesamten Modelllebensdauer können

Modelldynamik

Konzepte, Attribute und Relationen hinzugefügt werden. Schemamodifikationen sind also in einem Zustand möglich, in dem bereits Modelldaten existieren. Dazu siehe u. a. [Steinmann, 1997; Petzold, 2001; Willenbacher, 2002; Hauschild, 2003].

Dynamik unterstützende MVS Für die persistente Speicherung dynamischer Modelle sind spezielle Modellverwaltungssysteme (MVS) erforderlich, die das Hinzufügen oder Ändern von Elementen des Modellschemas zur Laufzeit ermöglichen. Zur Realisierung eines solchen Modellverwaltungssystems schlagen Willenbacher und Hauschild die explizite Repräsentation der Modellierungskonzeptebene als strukturierte Menge von Klassen vor, die jeweils ein Element des Modellierungskonzeptes (z. B. *Klasse*, *Attribut*, *Relation*, *Exemplar*) abbilden [Willenbacher, 2002; Hauschild, 2003]. Ausschließlich diese Metaklassen werden vom Modellverwaltungssystem implementiert. Die Abbildung von Modellschemata und Modelldaten erfolgt jeweils durch das Bilden von Exemplaren der Metaklassen. Abbildung 3.3 illustriert diesen Gesamtzusammenhang. Parallel zeigt Abbildung 3.4 die Struktur eines Modellverbundes nach dem Muster von Peer-DBMS, wobei jeweils ein Exemplar des Modellverwaltungssystems einen Peer realisiert.

Generischer Modellzugriff Das Modellverwaltungssystem stellt eine generische Schnittstelle¹⁴ zur Verfügung, über die sowohl Modellschemata als auch Modelldaten erzeugt, modifiziert und abgefragt werden können. Da dem Modellverwaltungssystem ausschließlich die Metaklassen bekannt sind und sowohl Modellschemata als auch Modelldaten als Ausprägungen der Metaklassen behandelt werden, sind keine speziellen Zugriffsmethoden auf einzelne Attribute einer Klasse möglich, wie beispielsweise `getOpenings()` für Exemplare einer Klasse `Wall`. Stattdessen müssen auf Metaebene definierte analysierende Methoden verwendet werden wie `getClass()` oder `getAttributeByName(String name)` für Exemplare von `Instance`. Dazu vgl. [Willenbacher,

¹⁴ Programmierschnittstelle

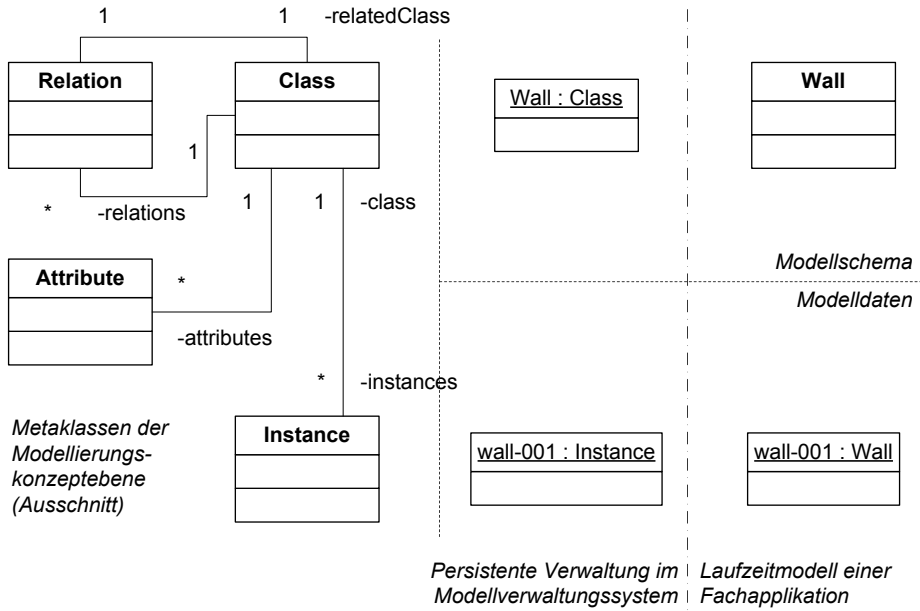


Abbildung 3.3.: Klassendiagramm der Ebenen dynamischer Modelle

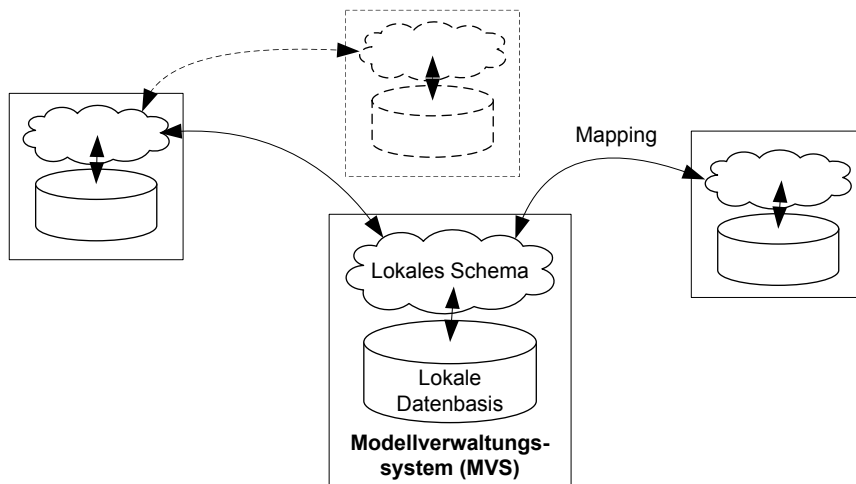


Abbildung 3.4.: Verbund einzelner Modellverwaltungssysteme

2002]. Auf der anderen Seite erlaubt das Modellverwaltungssystem mit Hilfe dieser Methoden den Zugriff auf Modelldatenobjekte als Ausprägungen der Metaklasse `Instance`, unabhängig davon, welcher Klasse und welchem Modellschema sie tatsächlich angehören. Die Autoren des dynamischen Modellierungansatzes bezeichnen diese Art des Modellzugriffs als *generischen Zugriff*, da er unabhängig von der Semantik eines konkreten Modellschemas ist.

Warum Modelldynamik? Als Begründung für diesen Ansatz wird von Hauschild in erster Linie die Forderung nach der Erweiterbarkeit und Flexibilität von Bauwerksmodellen genannt [Hauschild, 2003]. **Erweiterbarkeit** bedeutet in diesem Zusammenhang die Möglichkeit zur Erweiterung der Menge der Definitionen, die das konzeptionelle Schema des betrachteten Anwendungsbereiches, d.h. das Modellschema, bilden. **Flexibilität** steht für die Möglichkeit zur Definition neuer, spezifischer Informationseinheiten sowie das Hinzufügen und Anpassen von Relationen zwischen Informationseinheiten.

Als konkrete Anlässe, die eine Modifikation des Modellschemas bei bestehendem und fortzuführendem Datenbestand erforderlich machen, werden folgende Situationen genannt:

Vage und unscharfe Informationen **Bauwerksentwurf und Bauaufnahme.** Während dieser Arbeitsschritte sind oft vage oder unscharfe Informationen zu verwalten, die erst im Verlauf der Bearbeitung sukzessive konkretisiert werden. Art und Umfang der abzubildenden Informationen sind oft sehr individuell und ergeben sich jeweils aus der konkreten Situation. Die Vorgabe eines starren Systems zur Strukturierung der Informationen wirkt restriktiv und hindernd bei der Bearbeitung [Steinmann, 1997; Petzold, 2001].

Fach- oder regionalspezifische Informationen **Projektspezifische Informationen.** Standardisierte Modellschemata wie die IFC verzichten auf die Abbildung ausschließlich fach- oder regionalspezifischer Informationen. Die Repräsentation von Domänenwissen, wie

beispielsweise Regeln zum baulichen Brandschutz, ist ausschließlich auf Basis standardisierter Schemata nicht möglich [Hauschild, 2003].

Veränderungen während der Modelllebensdauer. Unter dem Anspruch einer lebenszyklusübergreifenden Bereitstellung der Modelldaten muss mit sehr langen Lebenszeiten der Modelle (= Schemata + Daten) gerechnet werden. Entwicklungen neuer Bautechnologien oder Nutzungsanforderungen, Änderungen technischer Normen und gesetzlicher Regelungen sind für diese Zeiträume nicht absehbar. Entwicklungen dieser Art stellen Veränderungen des betrachteten Anwendungsbereiches *Bauwesen* dar, die sich in Änderungen der Konzepte, Attribute und Beziehungen innerhalb des konzeptuellen Schemas niederschlagen. Während der Modelllebensdauer muss außerdem mit einem Wandel der bestehenden Softwarelandschaft gerechnet werden, der ebenso Einfluss auf die verwendeten, d.h. die durch die Applikationen interpretier- und verarbeitbaren Schemata hat. Dazu vgl. [Steinmann, 1997; Willenbacher, 2002; Hauschild, 2003].

*Entwicklung des
Domänenwissens*

Insgesamt betrachtet bringen dynamisierte Modelle jedoch eine Reihe eigener Probleme mit sich. Über die generische Zugriffsschnittstelle des Modellverwaltungssystems kann die spezifische Semantik der Modellschemata nicht transportiert werden. Das bedeutet, Applikationen, die auf ein im Modellverwaltungssystem vorgehaltenes Modell zugreifen wollen, können ausschließlich die Klassen, Attribute und Relationen abrufen, deren Bezeichnung sie „kennen“. Ihnen unbekannte Schemaelemente können die Applikationen weder vom Modellverwaltungssystem abrufen noch interpretieren und verarbeiten.

*Probleme der
Modelldynamik*

Die Bereitstellung spezifischer Verarbeitungslogik als Bestandteil der Modellschemata ist nicht vorgesehen. Folglich ist die Interpretier- und Verarbeitbarkeit der Modelldaten unmittelbar abhängig von Applikationen, die die entsprechenden Modellschemata implementieren und die zugehörige Verarbeitungslogik bereitstellen. Parallel zur Definition neuer Schema-

*Modelle stellen keine
Verarbeitungslogik
bereit*

elemente muss daher außerhalb des Modellverwaltungssystems für die Implementierung der zugehörigen Verarbeitungslogik gesorgt werden. Dieser Zusammenhang stellt kein Problem dar für Modifikationen, die sich aus Versionsänderungen von Fachapplikationen ergeben. In diesem Fall ist lediglich die Nachführung des Modellschemas im Modellverwaltungssystem notwendig. Der umgekehrte Fall stellt jedoch ein Problem dar, das an folgendem Beispiel illustriert werden soll (siehe auch Abbildung 3.5):

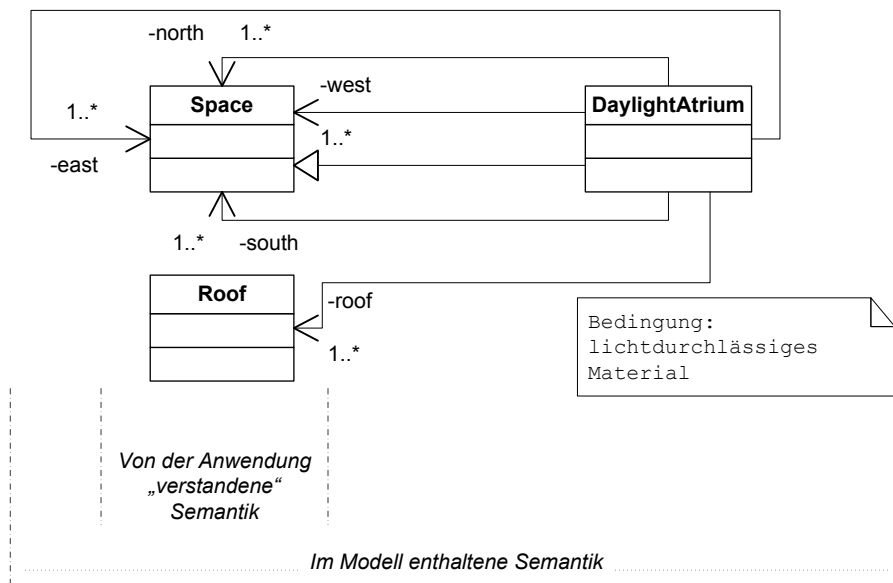


Abbildung 3.5.: Konflikt bei dynamischer Modellerweiterung

Beispiel **DaylightAtrium** Angenommen, während des Gebäudeentwurfs legt der Entwerfende über die Schnittstelle des Modellverwaltungssystems eine neue Klasse **DaylightAtrium** als Unterklasse einer bestehenden Klasse **Space** an. Der Entwerfende legt für die neue Klasse ferner vier Relationen zur Verknüpfung angrenzender Räume an (**east**, **west**, **north**, **south**) und legt fest, dass jede auf mindestens ein Exemplar von **Space** verweisen muss, da sich das Atrium stets im Inneren des Gebäudes befinden soll. Eine fünfte Relation **roof**

könnte auf ein Bauteil verweisen, dessen Material lichtdurchlässig sein soll. Eine aufsetzende Applikation, die die neue Klasse `DaylightAtrium` nicht kennt, kann die spezifischen Informationen von `DaylightAtrium`-Exemplaren nicht interpretieren. Die Exemplare können zwar als Exemplare der Oberklasse `Space` behandelt werden. Da die Restriktion der vierseitigen Begrenzung durch andere Räume für die Klasse `Space` jedoch nicht definiert ist, kann die Applikation nicht für die korrekte Behandlung eines `DaylightAtrium`-Exemplars sorgen und verhindern, dass es beispielsweise an eine Außenkante des Gebäudes verschoben wird. Diese Situation, in der zwar eine rechnerinterne Repräsentation eines Entwurfsobjektes realisiert ist, diese jedoch nicht computergestützt bearbeitet werden kann, stellt genau genommen einen Widerspruch zum grundsätzlichen Anliegen von Bauwerksmodellen dar: der digitalen Repräsentation bauwerksbezogener Informationen zum Zweck der computergestützten Bearbeitung.

Vor diesem Hintergrund wird in der vorliegenden Arbeit von der Verwendung dynamischer Modelle abgesehen. Den Forderungen nach Flexibilität und Erweiterbarkeit kann weitgehend mit anderen Mechanismen entsprochen werden. Spezifische regional oder fachbezogen variierende Informationen können durch den Austausch von Teilmodellen im Verbund integriert werden. Dieses Prinzip wurde bereits erwähnt und ist in [Rüppel u. a., 2006, 2007; Willenbacher, 2002] beschrieben.

*Verzicht auf
Modelldynamik*

Im Rahmen früher Entwurfsphasen ist vor allem die flexible Strukturierung der externalisierten Informationen zum Entwurfsgegenstand gefordert. Eine maschinengesteuerte Bearbeitung der frühen Entwurfsphasen ist nach derzeitigem Erkenntnisstand nicht realisierbar. Folglich wird das Erbringen der kreativen Entwurfsleistungen in absehbarer Zukunft stets eine von Menschen zu lösende Aufgabe bleiben. Die Strukturierung der Informationen dient dabei vorrangig dem Zugriff auf die Informationen durch Menschen, weniger der Interpretation und Verarbeitung durch Software.

*Flexibilität der
Informationsstrukturierung*

*Zusätzliche
Strukturierung
oberhalb der Modell-
verwaltungsebene* Eine von der tatsächlichen physischen Repräsentation der Informationen unabhängige Strukturierung mit Hilfe separat erstellter und verwalteter Verknüpfungen zwischen den Informationseinheiten ist für diesen Zweck wesentlich flexibler. Ein entsprechendes Konzept wird in Kapitel 7 dieser Arbeit beschrieben. Diese separat verwalteten Verknüpfungen haben keinen Einfluss auf die konzeptuellen Schemata der Bauwerksmodelle. Dadurch können einerseits verfügbare Softwarewerkzeuge ohne Einschränkung verwendet werden. Andererseits werden die Entwerfenden nicht mit der Problematik der Definition neuer Schemaelemente sowie zugehöriger Verarbeitungslogik konfrontiert. Über die separat verwaltete Verknüpfungsstruktur können Informationen unabhängig von ihrer physischen Repräsentation flexibel einzelnen Entwurfsobjekten zugeordnet werden, so dass eine Art flexibles Inhaltsverzeichnis für die Informationen zum Entwurfsgegenstand erstellt wird. Die Zuordnung kann bei Bedarf geändert werden, ohne die physische Repräsentation der Informationen ändern zu müssen.

3.8. KONSEQUENZEN FÜR EINE DIGITALE BAUWERKSDOKUMENTATION

Resultierend aus den in diesem Kapitel erläuterten Methoden und Techniken zur Verwaltung und Integration bauwerksbezogener Informationen ergeben sich für den Rahmen dieser Arbeit folgende Konsequenzen:

- IFC allein deckt
Bedarf nicht*
- Standardisierungsbemühungen wie die IFC zielen auf die Bereitstellung eines allgemein anerkannten globalen Bauwerksmodellschemas und leisten damit einen wesentlichen Beitrag zur Integration im Bauwesen. Eine vollständige Integration auf Basis eines globalen Schemas wird jedoch aufgrund der Spezifik des Bauwesens nicht möglich sein.

- Langfristig ist der Ansatz eines projektbezogen zusammengesetzten Modellverbundes nach dem Muster der Peer-DBMS am realistischsten. Ein das IFC-Schema implementierender Peer kann in einem solchen Modellverbund als zentraler Knoten agieren, mit dem alle anderen Peers bzw. Teilmodelle durch Mappings verknüpft werden. Aufgrund der extremen Heterogenität der Softwarelandschaft und der sich nur langsam etablierenden Neuerungen kann nicht für jedes Projekt ein IFC-basierter zentraler Peer als gegeben angenommen werden. Die Struktur eines solchen Modellverbundes zeigt Abbildung 3.6.

*Variabler
Modellverbund*

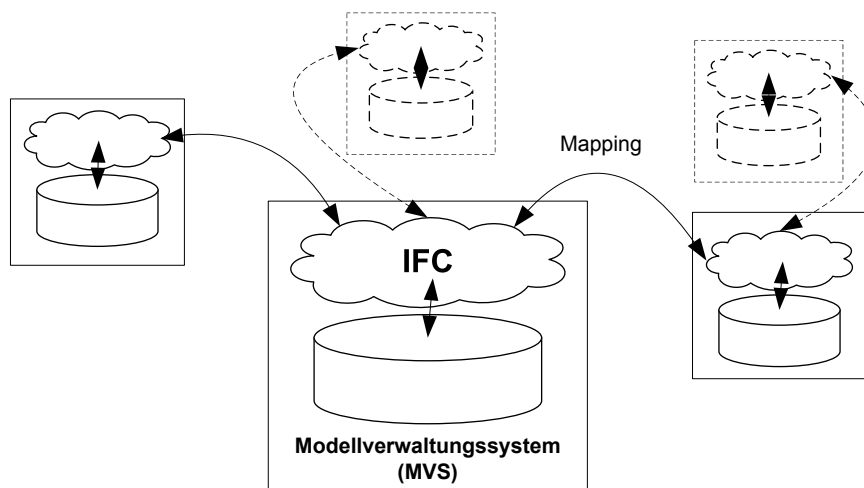


Abbildung 3.6.: Modellverbund mit zentralem IFC-Peer

- Eine vollständige Realisierung und Verwendung dynamischer Modelle ermöglicht eine maximale Flexibilität und Erweiterbarkeit der Modelle, bringt jedoch zusätzliche Probleme mit sich. Insbesondere bezüglich der Bereitstellung entsprechender Anwendungslogik sind diese nicht ohne Weiteres lösbar. Dynamik unterstützende Modellverwaltungssysteme, die auf Basis der Modellierungskonzeptebene arbeiten und entsprechende Metaklassen implementieren, bieten jedoch einen

*Statische
Modellschemata*

Vorteil. Modellklassen und deren Exemplare können unabhängig von ihrer konkreten Beschaffenheit und Zuordnung zu einem konzeptuellen Schema als Exemplare der Metaklassen behandelt werden. Auf diese Weise ist grundsätzlich immer ein informierender Zugriff auf die einzelnen Elemente möglich.

*Datenverwaltung der
Bauwerks-
dokumentation*

Somit kann eine digitale lebenszyklusübergreifende Bauwerksdokumentation sinnvoll nur auf der Basis eines variablen Modellverbundes realisiert werden. Eine solche Struktur ermöglicht einen dem spezifischen Bedarf des jeweiligen Projektes entsprechenden Zusammenschluss einzelner Peers. Nach Projektabschluss kann der Modellverbund in eine Bauwerksdokumentation übergehen und bei Bedarf im Verlauf der Zeit um weitere Peers ergänzt werden. Gleichzeitig hat eine solche Struktur aber auch einen entscheidenden Nachteil. Ihre Systematik spiegelt in erster Linie die Zusammensetzung eines Projektteams wider und nicht das Bauwerk als das eigentliche Objekt der Betrachtung. Die einzelnen Peers verwalten jeweils nur die für ihre fachspezifische Betrachtung relevanten Informationen zu den einzelnen Elementen¹⁵ des Bauwerks. Dies hat zur Folge, dass über den gesamten Modellverbund verteilt unterschiedliche Repräsentationen ein und desselben Elementes existieren können. Aufgrund des Fehlens eines zentralen Einstiegspunktes ist es dabei sehr schwer, alle zu einem Element gehörenden Repräsentationen zu finden.

Diese Problematik der Organisation und Strukturierung der Bauwerksdokumentation bildet eine der zentralen Fragestellungen im weiteren Verlauf dieser Arbeit. Zunächst aber sollen in den beiden folgenden Kapiteln die Anforderungen ermittelt werden, die aus der Sicht der entwurfsbezogenen Informationssuche an die Strukturierung des Datenbestandes und den Zugriff darauf gestellt werden.

¹⁵ d. h. bei Gebäuden den Räumen und Bauteilen

4. ENTWERFEN ALS KREATIVES PROBLEMLÖSEN

Im vorangegangenen Kapitel wurden die technischen Grundlagen zur Verwaltung einer digitalen Bauwerksdokumentation erläutert. Dieses und das nächste Kapitel widmen sich den Anforderungen an die Nutzerschnittstelle zur Informationssuche in einer solchen Bauwerksdokumentation. Dabei soll anhand des Fragekomplexes b) (siehe S. 11) herausgearbeitet werden, wovon die Leistungsfähigkeit eines Entwerfenden abhängt und wodurch sie beeinflusst wird. Folgende Fragen sind in diesem Zusammenhang zu beantworten: Wie gehen Entwerfende bei der Lösung eines Entwurfsproblems vor? Welche, vor allem gedanklichen, Leistungen müssen sie dabei erbringen? Wie werden diese erbracht?

Ausgangspunkt

Inhalt

4.1. ABGRENZUNG

Charakteristisch für Bauprojekte ist die jeweils einmalige Konstellation von Anforderungen, Standort und beteiligten Personen seitens der Planer wie auch der Auftraggeber [Hübler u. a., 2003; Donath, 2003; von Both, 2004]. Ebenso einzigartig ist folglich auch der jeweilige Prozess der Lösungsfindung. Entwerfen ist eben dieses Finden der Lösung und gleichzeitig die Entwicklung von externen Repräsentationen, die anderen Beteiligten die Idee der Lösung vermitteln können.

*Einmaligkeit von
Entwurfsaufgaben*

4. ENTWERFEN ALS KREATIVES PROBLEMLÖSEN

Klassifikation von Entwurfsproblemen I

Nach [Rowe, 1987; Brown u. Chandrasekaran, 1989; Karbach u. Linster, 1990] können Entwurfs- und Planungsprobleme hinsichtlich ihrer Ansprüche an die zu entwickelnde Lösung drei verschiedenen Klassen zugeordnet werden, wie Tabelle 4.1 zeigt.

<i>Variantenkonstruktion</i>	<i>Änderungskonstruktion</i>	<i>Neukonstruktion</i>
Class 3 Design [Brown u. Chandrasekaran, 1989]	Class 2 Design	Class 1 Design
Routine Design [Karbach u. Linster, 1990]	Innovative Design	Creative Design
Well-defined Problem [Rowe, 1987]	Ill-defined Problem	Wicked Problem
Bekannte Anforderungen	Neue Anforderungen	Neue Anforderungen
Vorhandene (bekannte) Lösungselemente	Vorhandene (bekannte) Lösungselemente	Nicht vorhandene (unbekannte) Lösungselemente
Bekannte Elementkombinationen	Neue Elementkombinationen	Neue Elementkombinationen

Tabelle 4.1.: Klassen von Entwurfsproblemen

Unterstützbarkeit durch Softwaresysteme

Anhand dieser Klassifikation können Aussagen hinsichtlich der Unterstützbarkeit der Lösungsfindung durch Softwaresysteme getroffen werden [Steinmann, 1997]. Eine vollständige Automatisierung bzw. das Generieren von Lösungsvorschlägen ist nur für ausgewählte Probleme der Variantenkonstruktion möglich. Dafür können die in Abschnitt 2.5 erwähnten Entwurfsgeneratoren eingesetzt werden. Voraussetzung ist dabei jedoch eine vollständige Beschreibung des Problems sowie der Regeln zur Lösungsfindung. Für Probleme der Änderungskonstruktion bieten sich passive, modellbasierte Werkzeuge an, die eine interne Repräsentation der verfügbaren

Lösungselemente bereitstellen, wie beispielsweise CAAD-Systeme. Probleme der Neukonstruktion sind jedoch nur indirekt unterstützbar, indem den Entwerfenden Werkzeuge zur Erstellung externalisierter Repräsentationen der Entwurfslösung bzw. zur systematischen Verwaltung von Informationen zur Entwurfslösung zur Verfügung gestellt werden.

Planungsaufgaben im Kontext bestehender Bausubstanz können ebenfalls anhand dieser drei Klassen unterschieden werden. Maßnahmen der Instandhaltung und Modernisierung bestehender Bauwerke durch Austausch defekter Bauteile oder Ergänzung von Ausbauelementen (z. B. Wärmedämmung) oder Installationen (Heizung) sind der Klasse Variantenkonstruktion zuzuordnen. Werkzeuge zur Unterstützung dieser Aufgaben wurden u. a. im Rahmen von [GISMO Projekt, 2008] entwickelt. Projekte, die jedoch eine Umnutzung, bauliche Erweiterung oder den Umbau der bestehenden Substanz vorsehen, müssen der Änderungskonstruktion bzw. Neukonstruktion zugerechnet werden.

*Aufgabenstellungen
der Planung im
Bestand*

Besonders sensibel sind bei derartigen Projekten die frühen Phasen (Vorplanung/Vorstufe, vgl. S. 24), in denen das Gesamtkonzept für das Projekt entwickelt wird. Ein wesentlicher Anteil dieser Phasen besteht in der Erfassung und Definition der Problemstellung im Dialog mit dem Bauherrn und durch Analyse des bestehenden Bauwerks. Für derartige Problemstellungen existieren bislang kaum unterstützende Werkzeuge [Donath u. Petzold, 2005]. Werkzeuge zur systematischen Erfassung und Verwaltung planungsrelevanter Informationen sind allein zu diesem Zweck nicht ausreichend. Den Entwerfenden muss zugleich das gezielte, problembezogene Wiederfinden von Informationen ermöglicht werden. Eben diesem Zweck soll die zu entwickelnde Informationsumgebung dienen.

*Mangelnde
Unterstützung früher
Phasen*

Vor diesem Hintergrund wird Entwerfen im Folgenden als eine kreativ-schöpferische Tätigkeit betrachtet, die in erster Linie eine menschliche Verstandesleistung darstellt. Derartige Verstandesleistungen werden in einer

*Entwerfen als
Verstandesleistung*

Teildisziplin der Psychologie unter der Bezeichnung *kreatives Problemlösen* untersucht.

Allerdings kann Kreativität nicht kausal hervorgebracht werden [Brodbeck, 1995]. Das heißt, das „Entstehen“ von Entwurfslösungen ist nicht vollständig rational fassbar, wie u. a. folgende Zitate zeigen:

„Tätigkeiten und Entwurfsstrategien wechseln sich im Verlauf des Entwurfsprozesses nichtvorhersagbar ab.“ [Steinmann, 1997, S. 53]

„Entwurfsideen werden nach entsprechender gedanklicher Arbeit oft spontan geboren und in spontanen Skizzen festgehalten.“ [Lorenz, 2004, S. 139]

„Methodik ist kein Ersatz für Intuition.“ [Joedicke, 1976, S. 8]

„Der rationale Akt der Auswahl zwischen gegebenen Alternativen setzt die transrationale Entdeckung dieser Alternativen voraus.“ [Brodbeck, 1995, S. 14]

„Planen ist [...] nicht als ein rein rationaler Prozess zu verstehen, sondern stellt auch eine schöpferische Arbeit dar, die weitgehend auf den Fähigkeiten, Erfahrungen und der Intuition des Menschen beruht. Dies gilt besonders für die Bauplanung, bei der vor allem die Entwurfsprozesse auf die kreativen Leistungen des menschlichen Geistes angewiesen sind.“ [von Both, 2004, S. 1]

*Untersuchung
gedanklicher
Aktivitäten* Vor diesem Hintergrund zielt dieses Kapitel nicht auf eine formale Beschreibung von Entwurfsstrategien. Stattdessen sollen einerseits die Charakteristika gedanklicher Abläufe bzw. Aktivitäten beim Lösen verschiedener Arten von Problemen herausgearbeitet werden. Andererseits sollen – soweit möglich – Faktoren identifiziert werden, die die „kreative Leistungsfähigkeit“ eines Entwerfenden, insbesondere beim Finden von Lösungsalternativen, beeinflussen.

4.2. BEGRIFFE

In der Denkpsychologie wird ein Problem allgemein durch die drei folgenden Komponenten beschrieben: Ein **unerwünschter Ausgangszustand** soll in einen **gewünschten Zielzustand** überführt werden, wobei eine **Barriere** zwischen beiden Zuständen existiert, die nicht ohne kognitive (intellektuelle) Anstrengung überwunden werden kann. Dazu vgl. [Dörner, 1976; Krech u. a., 1992; Holzinger, 2001].

Problem

Problemlösen wird nach [Krech u. a., 1992, Bd. 4] dann als kreativ bezeichnet, wenn es für das betreffende Individuum neu ist, unabhängig davon, ob die spezielle Lösung zuvor bereits von einem anderen Individuum produziert wurde. Diese Sichtweise ist jedoch sehr eng gefasst. Brodbeck geht davon aus, dass Kreativität weder objektivierend beschrieben noch gemessen werden kann, sondern situationsbezogen ist [Brodbeck, 1995]. Von der Bedeutung, die eine Sache für ein Individuum oder eine Gruppe in einer bestimmten Situation hat, hängt demnach ab, ob das Hervorbringen dieser Sache als kreativer Prozess bezeichnet werden kann [Boden, 1994]. Kreative Lösungen sind somit eng an die individuelle Wahrnehmung einer gegebenen Situation gebunden. In diesem Sinne hat das Erfassen des Problems und seines Kontextes sowie das Sammeln von für die Lösung relevanten Informationen einen wesentlichen Anteil am Prozess der Lösungsfindung. Die Art der entwickelten Lösung ist untrennbar damit verbunden, wie der Entwerfende das Problem wahrnimmt, definiert und artikuliert [Ching, 1996].

*Kreativität und
Situationsbezug*

4.3. ZUM VORGANG DES KREATIVEN PROBLEMLÖSENS

Drei Phasen der Iteration Problemlösevorgänge finden typischerweise in einer iterativen Wiederholung unterschiedlicher gedanklicher Aktivitäten statt. Prinzipiell können dabei drei verschiedene Arten solcher Aktivitäten bzw. Phasen eines Iterationszyklus unterschieden werden. Verschiedene Autoren schlagen darüber hinaus detailliertere Gliederungen vor [Marchionini, 1997; Grüning u. Kühn, 2004], auf die hier jedoch nicht näher eingegangen werden soll.

Zu den folgenden Erläuterungen vgl. u. a. [Dörner, 1976; Rowe, 1987; Krech u. a., 1992; Marchionini, 1997; Gero, 1999; Grüning u. Kühn, 2004].

4.3.1. PHASEN DES ITERATIONSPROZESSES

Analyse In der ersten **analysierenden** Phase werden relevante Informationen zum Problem zusammengetragen und systematisiert, Zielsetzungen werden formuliert und das Problem in brauchbarer Weise repräsentiert. Die Psychologie unterscheidet dabei nicht zwischen mentalen und externalisierten Repräsentationen. Das Erstellen externer Repräsentationen ist ohnehin erst dann möglich, wenn bereits mentale Modelle entwickelt wurden. In Abhängigkeit vom Grad der Strukturiertheit des Problems kann in dieser Phase auch eine systematische Planung der anzuwendenden Lösungsmethode stattfinden.

Synthese Die sich anschließende **synthetisierende** Phase bringt Lösungsansätze, gegebenenfalls verschiedene Lösungsalternativen hervor. Bislang existiert jedoch nur ein sehr begrenztes Verständnis der in dieser Phase ablaufenden gedanklichen Aktivitäten [Krech u. a., 1992, Bd.4]. Durch empirische Untersuchungen konnten verschiedene Handlungsstrategien identifiziert werden, die das Finden von Lösungen unterstützen. Auf diese wird im folgenden Ab-

schnitt eingegangen. Die Frage, was das Hervorbringen von Lösungsideen letztlich bewirkt und wie diese entstehen, ist jedoch weitgehend ungeklärt. Krech u. a. sprechen bildhaft von einem „Aha!“-Erlebnis bzw. einem Licht, das einem aufgeht [Krech u. a., 1992]. Dörner verwendet die Bezeichnung „plötzlicher Einfall“ [Dörner, 1976]. Lorenz beschreibt spontan entstehende Skizzen [Lorenz, 2004]. Diese Phase stellt somit einen sich jeglicher Formalisierung entziehenden, zugleich aber zentralen Bestandteil des Problemlösens dar.

*Synthetisierende
Tätigkeit ist nicht
formalisierbar!*

Schließlich müssen die gefundenen Lösungsideen in einer **evaluierenden** und **reflektierenden** Phase auf ihre Tragfähigkeit und Entsprechung zur Zielsetzung untersucht werden. Dabei werden die gefundenen Lösungen getestet, gegebenenfalls Alternativen ausgewählt. Wurde keine zufriedenstellende Lösung erreicht, erfolgt ein Rücksprung auf eine der beiden vorherigen Phasen.

Evaluierung

Untersuchungen des Problemlösens fanden und finden unter verschiedenen Gesichtspunkten und nicht nur als Teildisziplin der Kognitionspsychologie statt. Iterative Vorgehensweisen mit einem Wechsel der drei Phasen werden von verschiedenen Autoren in unterschiedlichen Kontexten beschrieben. Tabelle 4.2 stellt die von den verschiedenen Autoren verwendeten Bezeichnungen der drei Phasen gegenüber. Die Ausführungen in [Krech u. a., 1992] beziehen sich dabei auf Untersuchungen, die in den 20er Jahren von Graham Wallas durchgeführt wurden und die auf das Lösen formaler (*well-defined*) Probleme fokussieren. Wallas beschrieb zwei Abschnitte der synthetisierenden Phase: die *Inkubation* als bewusstes Beiseitelegen des Problems nach der Vorbereitungsphase und die *Erleuchtung* als plötzlich eintretendes „Aha!“-Erlebnis nach einem gewissen Zeitraum. Ähnliches beobachtete Lorenz (vgl. Zitat auf S. 72). In [Zimmermann u. Campillo, 2003] wird das problemorientierte, selbstgesteuerte Lernen als eine spezielle Form des Problemlösens betrachtet. Gero bezieht sich auf Studien zur Vorgehensweise menschlicher

*Bezeichnungen der
drei Phasen*

*Inkubation und
Erleuchtung*

4. ENTWERFEN ALS KREATIVES PROBLEMLÖSEN

	Analysierende Phase	Synthetisierende Phase	Evaluierende, reflektierende Phase
<i>Klassisches Problemlösen</i> [Krech u. a., 1992]	Vorbereitung	Inkubation + Erleuchtung	Verifizierung
<i>Problemorientiertes Lernen</i> [Zimmermann u. Campillo, 2003]	Forethought phase	Performance phase	Self-reflection
<i>Entwerfen und Gestalten</i> [Gero, 1999]	Formulation	Synthesis	Evaluation
<i>Architekturentwurf</i> [Lorenz, 2004]	Informationsphase	Kreative Phase	Bewertungsphase
<i>Informationssuche</i> [Marchionini, 1997]	Verstehen des Problems	Planen und Ausführen der Suche	Bewertung und Verwendung der Ergebnisse

Tabelle 4.2.: Phasen des Problemlösevorgangs bei verschiedenen Autoren

Entwerfer und Gestalter [Gero, 1999], Lorenz untersuchte die Vorgehensweise verschiedener Architekten beim Entwerfen [Lorenz, 2004] und Marchionini beschreibt den Vorgang der Informationssuche [Marchionini, 1997].

In den 60er Jahren entwickelte Morris Asimov ein Modell zur Beschreibung von Entwurfs- und Gestaltungsprozessen, das den Iterationsprozess als Spiralbewegung mit gleichzeitiger Konkretisierung des Entwurfsgegenstandes beschreibt [Rowe, 1987; Gero, 1999]. Abbildung 4.1 zeigt die von Asimov vorgeschlagene Entwurfsspirale unter Berücksichtigung der beiden Entwicklungsrichtungen *Konkretisierung* und *Externalisierung*. Konkretisierung steht dabei für die fortschreitende Verfeinerung der Entwurfslösung vom Groben zum Detail, wie sie beispielsweise durch die neun Leistungsphasen der Honorarordnung für Architekten und Ingenieure beschrieben wird [HOAI, 1996]. Unter Externalisierung wird die Überführung mentaler Repräsentationen der Entwurfslösung in alle Arten von Entwurfsdokumentationen, wie Skizzen, physische 3D-Modelle oder Planzeichnungen verstanden. Abschnitt 4.5 geht auf die wechselseitigen Beziehungen zwischen mentalen und externalisierten Repräsentationen noch detaillierter ein.

„Entwurfsspirale“

*Konkretisierung und
Externalisierung*

Das von Asimov entwickelte Spiralmodell wurde anhand empirischer Untersuchungen von Gero erweitert [Gero, 1999]. Gero dokumentiert, dass Entwerfer nicht zwingend bei jedem Iterationsschritt alle drei Phasen durchlaufen, sondern dass auch Rücksprünge zwischen evaluierender und synthetisierender Phase zu beobachten sind. Diese werden mit dem Fortschreiten des Entwurfs häufiger. Rücksprünge zwischen evaluierender und analysierender Phase sind charakteristisch für den Beginn des Entwurfsvorgangs, werden dann aber zunehmend seltener. Abbildung 4.2 zeigt die möglichen Übergänge zwischen den Phasen.

*Erweiterung der
Spirale*

Prinzipiell ist auch der Übergang von der synthetisierenden zur analysierenden Phase denkbar, wenn der Entwerfende während der synthetisierenden Phase feststellt, dass zur Lösungsfindung relevante Informationen

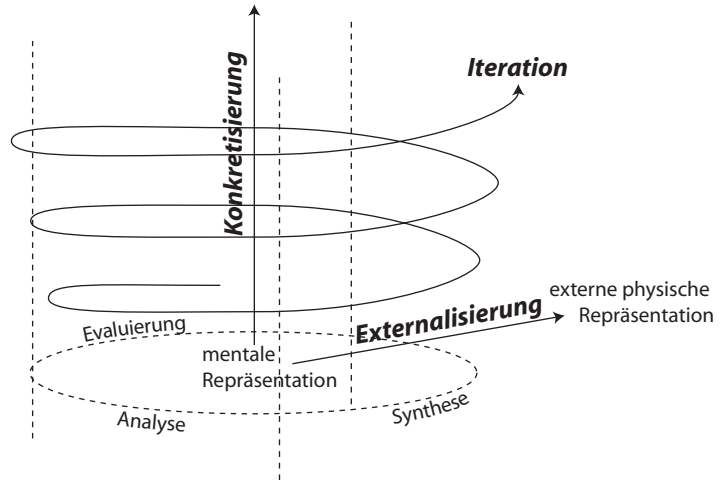


Abbildung 4.1.: Entwurfsspirale nach Asimov, erweitert um die Externalisierung mentaler Modelle

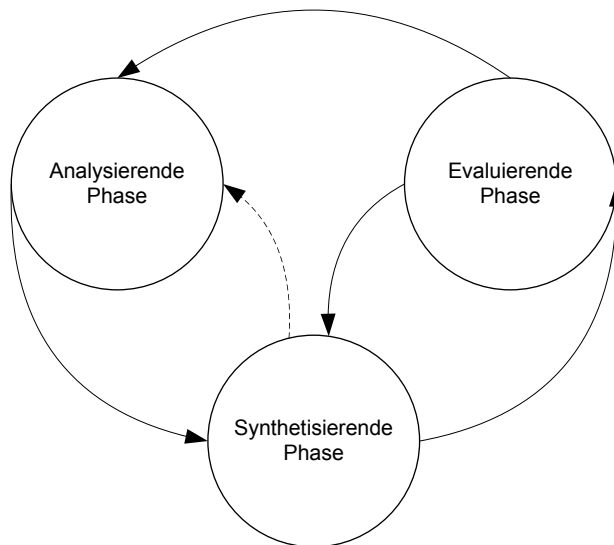


Abbildung 4.2.: Iterationsschleifen im Entwurfsprozess

fehlen. Diesbezüglich liegen jedoch keine Aussagen aus Untersuchungsergebnissen vor. Diese Betrachtung soll hier auch nicht weiter vertieft werden.

Den einzelnen Phasen des Iterationsprozesses können jeweils verschiedene Typen zu lösender Probleme zugeordnet werden:

- **Analysierende Phase**
Problem **Informationssuche**: Informationsdefizite müssen erkannt, relevante, aber bislang fehlende Informationen müssen gesucht und zusammengetragen werden.
- **Synthetisierende Phase**
Problem **Finden von Lösungsmöglichkeiten**: Lösungsideen sind zu „produzieren“ und zu ausgereiften Lösungsalternativen weiterzuentwickeln.
- **Evaluierende Phase**
Problem **Entscheidungsfindung**: Lösungsalternativen müssen beurteilt und potentielle Konsequenzen abgeschätzt werden. Auf dieser Basis muss ggf. aus den produzierten Lösungsalternativen eine zur weiteren Bearbeitung ausgewählt werden.

Der im zweiten Teil dieser Arbeit entwickelte Lösungsansatz fokussiert in erster Linie auf die Unterstützung der Informationssuche. Wie gezeigt wurde, ist diese integraler Bestandteil des hinsichtlich seines Verlaufes nicht vorhersehbaren Entwurfsprozesses. Daher muss davon ausgegangen werden, dass Informationsprobleme im Kontext des Entwerfens hinsichtlich des *Zeitpunktes* ihres Auftretens, hinsichtlich ihres *Gegenstandes* und hinsichtlich der geforderten *Granularität* der Antwort nicht vollständig bestimmbar sind. Dies bedeutet, dass eine automatisierte Auswahl und Präsentation der benötigten Informationen aus der Bauwerksdokumentation nicht möglich ist. Ein Entwerfender muss somit in die Lage versetzt werden, selbst interaktiv nach benötigten Informationen zu suchen.

*Konzentration auf
Unterstützung der
Informationssuche*

4.3.2. VARIANTEN DES ITERATIONSPROZESSES ENTSPRECHEND VERSCHIEDENER BARRIERETYPEN

Klassifikation von Entwurfsproblemen II: nach Arten von Barrieren

In Abhängigkeit von der Art des zu lösenden Problems kann der Iterationsprozess zur Lösungsfindung variieren. Dörner beschreibt drei verschiedene Arten von Barrieren, die ein Problem charakterisieren, und entsprechend drei Varianten iterativer Vorgehensweisen bei der Problemlösung [Dörner, 1976]. Anders als die häufig verwendete einfache Einteilung in *gut* und *schlecht strukturierte* Probleme [Pretz u. a., 2003; Zimmermann u. Campillo, 2003; Grüning u. Kühn, 2004] basiert die von Dörner vorgeschlagene Klassifikation von Problemtypen auf den zwei Dimensionen *Klarheit der Zielkriterien* und *Bekanntheitsgrad der Mittel* zur Transformation zwischen Ausgangs- und Zielzustand. Daraus ergibt sich die in Abbildung 4.3 dargestellte Matrix, wobei die Werte *hoch* und *gering* jeweils als Extreme einer Skala zu sehen sind, d. h. zwischen den Problemklassen gibt es keine scharfen Grenzen.

Problemklassifikation nach [Dörner 1976, S. 14]	Klarheit der Zielkriterien	
	<i>hoch</i>	<i>gering</i>
<i>hoch</i>	Interpolationsbarriere	Dialektische Barriere
Bekanntheitsgrad der Mittel		
<i>gering</i>	Synthesebarriere	Dialektische und Synthesebarriere

Abbildung 4.3.: Klassifikation von Barrieretypen nach [Dörner, 1976]

Alle Handlungsschritte, die in einer bestimmten Situation (tatsächlich oder gedanklich) ausgeführt werden können, um den gegebenen Zustand in den angestrebten Zielzustand oder ein Zwischenziel zu überführen, bezeichnet Dörner allgemein als *Operatoren*, die Gesamtheit aller in einer Situation möglichen Handlungsschritte als *Operatorinventar*. Diese Bezeichnungen werden auch in den folgenden Abschnitten in dieser Bedeutung verwendet.

*Operatoren und
Operatorinventar*

INTERPOLATIONSPROBLEME

Für Probleme, die durch klar formulierte Zielkriterien und ein vollständig bekanntes Operatorinventar gekennzeichnet sind, verwendet Dörner die Bezeichnung *Interpolationsproblem*. Die Barriere besteht in diesem Fall darin, die richtige Auswahl und Verkettung von Operatoren zu finden, wobei die Anzahl möglicher Kombinationen oft einen vollständigen Test derselben nicht zulässt.

*Auswahl und
Verkettung von
Operatoren*

Typische Interpolationsprobleme sind beispielsweise Brettspiele oder Kreuzworträtsel.

Beim Lösen solcher Probleme werden in der Regel die folgenden Aktivitäten ausgeführt [Dörner, 1976, S. 59ff]. Bestehende Abhängigkeiten zwischen den Schritten zeigt Abbildung 4.4.

Lösungsweg

1. Feststellen von Eigenschaften eines Zustandes oder Sachverhalts
(Wir benötigen ein Auditorium mit 180 Sitzplätzen.)
2. Vergleichen von Zuständen oder Sachverhalten
(Im bestehenden Gebäude ist keiner der Räume dafür groß genug.)
3. Suche nach Operatoren zur Beseitigung von Unterschieden
(Wir könnten das bestehende Gebäude so umbauen, dass Raum geschaffen wird.)

4. Bilden von Zwischenzielen

(Dafür müssen bestehende Funktionen temporär ausgelagert werden. Und wir müssen einen geeigneten Platz für das Auditorium innerhalb des Gebäudes finden.)

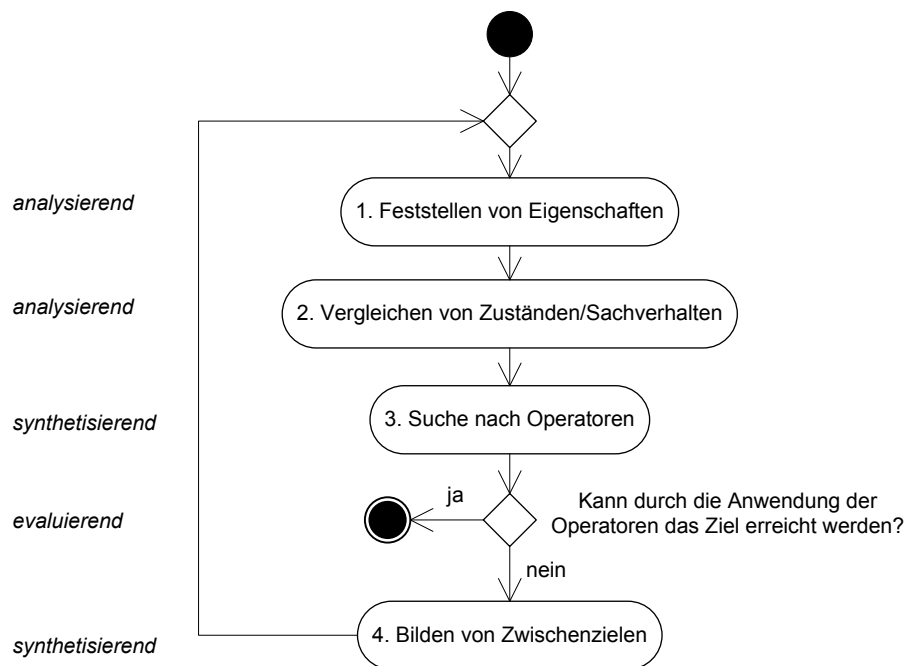


Abbildung 4.4.: Aktivitätsdiagramm: Ablauf beim Lösen von Interpolationsproblemen

Iteration und Suchrichtung In Abhängigkeit von der Anzahl der gebildeten Zwischenziele wiederholen sich diese Aktivitäten, wobei auf Schritt 4 wiederum Schritt 1 folgt. Eine Entwicklung des Lösungsweges kann sowohl vom Ausgangszustand (Vorwärtssuche) als auch vom Zielzustand aus (Rückwärtssuche) oder aber durch eine Kombination beider Strategien stattfinden.

SYNTHESEPROBLEME

Bei einem klar formulierten Zielzustand, jedoch einem unbekanntem oder unvollständig bekannten Operatorinventar spricht Dörner von einem *Syntheseproblem* [Dörner, 1976, S. 82ff].

*Finden bzw.
Konstruieren von
Operatoren*

Medizinische oder allgemein naturwissenschaftliche Forschung ist dieser Art von Problemen zuzurechnen.

Die Herausforderung für den Problemlöser besteht in diesem Fall im Finden oder Konstruieren der geeigneten Operatoren zur Transformation in den Zielzustand. Eben die Unbekanntheit der in Frage kommenden Operatoren ist ein wesentlicher Faktor, der die Problemlösung behindert. Ein weiterer Faktor sind eventuelle Fehleinstellungen des Problemlösers bzw. Entwerfenden bezüglich des Problems, die die Menge der in Betracht gezogenen Operatoren ungerechtfertigt einschränken. Letzterem kann durch ein bewusstes gedankliches Umstrukturieren des Problems entgegengewirkt werden, was jedoch flexible mentale Repräsentationen der gegebenen Situation voraussetzt. Dazu siehe auch Abschnitt 4.4.

In beiden Fällen ist die Suche nach Analogien eine hilfreiche Arbeitsstrategie, die von einer Person mehr oder weniger gezielt eingesetzt werden kann. Die Vorgehensweise wird im Folgenden beschrieben, wobei Abbildung 4.5 zeitliche bzw. logische Abhängigkeiten zwischen den Teilschritten illustriert.

Lösungsweg

1. Abstraktion des betrachteten Sachverhalts
(Die goldene Krone wird als ein „kompliziert berandeter Materieklumpen“ mit Volumen X betrachtet)
2. Suche nach einer alternativen Konkretisierung
(Auch Archimedes ist ein „kompliziert berandeter Materieklumpen“ mit Volumen Y.)

3. Übertragung von Merkmalen aus der gefundenen Konkretisierung
(Archimedes verdrängt entsprechend seines Volumens Y Wasser aus dem Badezuber. Die Krone muss also Wasser entsprechend ihres Volumens X verdrängen.)
4. Prüfen der Gültigkeit der übertragenen Merkmale
(Volumenbestimmung für die Krone)

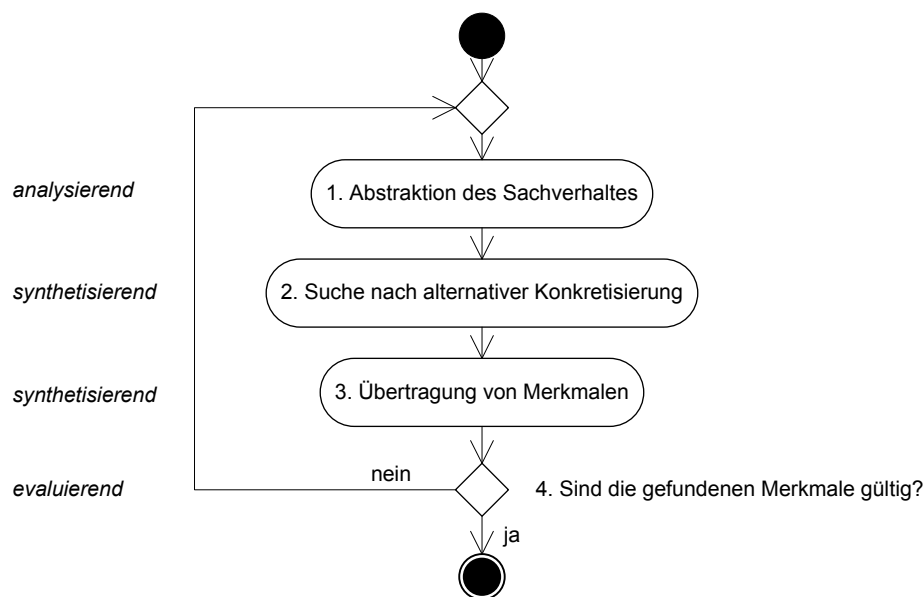


Abbildung 4.5.: Aktivitätsdiagramm: Ablauf beim Lösen von Syntheseproblemen mit Hilfe von Analogien

Iteration Die einzelnen Schritte werden wiederum zyklisch durchlaufen, so lange, bis eine Lösung des Problems in einer passenden Analogie gefunden wurde, d. h. ist Schritt 4 nicht erfolgreich, wird der Vorgang wiederholt, beginnend bei Schritt 1.

Plötzliche Einfälle Ein entscheidendes, bislang jedoch nicht näher erklärtes und auch nicht bewusst beeinflussbares Phänomen bei der Lösung von Syntheseproblemen, genauer beim Finden der geeigneten Operatoren, ist das, was im Allgemei-

nen als *plötzlicher Einfall* bezeichnet wird. Für dieses Phänomen konnte bisher nur eine Reihe begünstigender Faktoren ermittelt werden, die sein tatsächlichen Eintreten wahrscheinlicher werden lassen, jedoch keineswegs garantieren können. Auf diese Faktoren geht Abschnitt 4.4 näher ein.

DIALEKTISCHE PROBLEME

Ist zwar das verfügbare Operatorinventar bekannt, der zu erreichende Zielzustand jedoch nur vage formuliert, liegt nach Dörner ein *dialektisches Problem* vor. Urlaubsplanung oder Berufswahl können dieser Problemklasse zugeordnet werden. Die Überwindung der Barriere erfordert das – zumeist gedankliche – Festlegen und Prüfen möglicher Zielzustände. Charakteristisch ist dabei ein iterativer Prozess der Lösungswahl, Lösungsbewertung und Problemredefinition, so lange, bis ein befriedigender Zustand erreicht ist [Dörner, 1976, S. 95ff].

*Setzen und
Modifizieren von
Zielzuständen*

Abbildung 4.6 zeigt die zeitliche bzw. logische Abfolge der erforderlichen Teilschritte. Im Einzelnen sind dies:

Lösungsweg

1. (meist willkürliches) Setzen eines Zielzustandes
(Wir benötigen ein Auditorium. Legen wir es in den nördlichen Bereich des Gebäudes und entfernen wir dafür die Zwischendecke zwischen Erdgeschoss und erstem Obergeschoss an dieser Stelle!)
2. Test: Ist dieser Zielzustand befriedigend?
(Diese Idee muss konkretisiert werden!)
3. Ermitteln der inneren und äußeren Zwänge der Situation
(Wir müssen zwei Rettungswege haben. Außerdem sollen im Erdgeschoss verschiedene andere öffentliche Funktionen realisiert werden.)
4. Prüfen der einzelnen Teile des Zielzustandes auf Konsistenz mit den ermittelten Zwängen

4. ENTWERFEN ALS KREATIVES PROBLEMLÖSEN

(Die Rettungswege sind kein Problem, da Ausgänge zu ebener Erde möglich sind. Aber im Erdgeschoss ist dann nicht mehr genügend Platz für die übrigen öffentlichen Funktionen.)

5. Entfernen inkonsistenter Teile und Hinzufügen konsistenter Teile
(Verlagern wir einige der öffentlichen Funktionen in das erste Obergeschoss!)

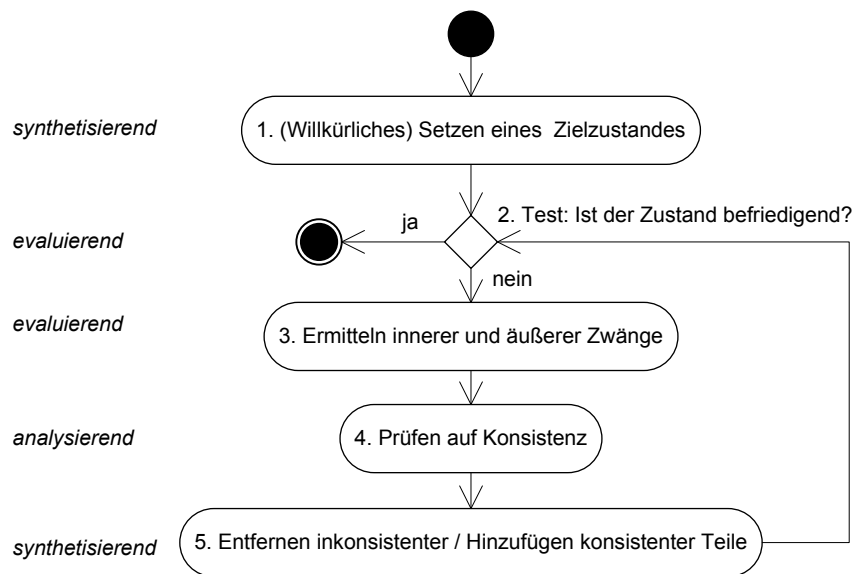


Abbildung 4.6.: Aktivitätsdiagramm: Ablauf beim Lösen dialektischer Probleme

Iteration Auf Schritt 5 folgt erneut der Test in Schritt 2 und in Abhängigkeit von dessen Ergebnis ein erneuter Durchlauf bzw. Abbruch des Vorgangs.

Parallele Entwicklung von Lösung und Zielkriterien Aus den in Schritt 3 ermittelten Zwängen ergeben sich bei jedem Iterationszyklus weitere Kriterien zur Bewertung der bis dahin gefundenen Lösung. Das bedeutet, die Kriterien zur Beurteilung des Zielzustandes werden *parallel* zum Prozess der Lösungsfindung entwickelt und verfeinert. Indem bereits gefundene Elemente der Gesamtlösung Randbedingungen erzeugen

oder ergänzen, bedingen sie das Hinzufügen weiterer Elemente zu einem stimmigen Ganzen.

Zum Beispiel bedingt die Anordnung der Haupterschließungsrichtungen auf dem Grundstück bzw. im Lageplan die innere Organisation des Gebäudes. Die Einordnung des Auditoriums bzgl. seiner Lage innerhalb des Gebäudes bedingt die Anlage von Flucht- und Rettungswegen. Usw.

Beispiel

Entwerfen bedeutet grundsätzlich den Umgang mit komplexen Problemen, für die Zielkriterien wie auch Mittel zur Lösung nur unvollständig bestimmt sind. Zudem muss eine Reihe nicht quantifizierbarer Aspekte in die Lösung einbezogen werden. In realen Problemstellungen überlagern sich meist mehrere Barrieren verschiedenen Typs [Dörner, 1976]. Insbesondere bei Aufgabenstellungen mit erhöhten künstlerischen oder technischen Ansprüchen sind weder Operatorinventar noch Zielkriterien vollständig bestimmbar. Meist lassen sich dabei immer wieder neue Aspekte finden, die eine Reformulierung des Problems bedingen [Rowe, 1987]. Somit greifen beim Entwerfen auch die Vorgehensweisen zur Lösungsfindung ineinander, das Setzen und Modifizieren von Zielzuständen beispielsweise erfolgt oft unter Zuhilfenahme von Analogien.

*Überlagerung von
Barrieren in realen
Entwurfsproblemen*

4.4. EINFLUSSFAKTOREN BEIM PROBLEMLÖSEN

Das Finden einer Lösung ist von verschiedenen Randbedingungen abhängig, die sich zum Teil der bewussten Kontrolle entziehen und nur indirekt beeinflusst werden können. Eines der zentralen Elemente ist dabei das bereits erwähnte Phänomen der spontanen Einfälle. Aus empirischen Untersuchungen sind verschiedene Faktoren bekannt, die die Entstehung spontaner Einfälle sowie das Finden eines geeigneten Lösungsansatzes begünstigen können. Die Erkenntnisse sind für einen erfahrenen Architekten gewiss nicht neu, wer-

den jedoch anhand der empirischen Beweisführung durch die Psychologie in konkreter Form fassbar. Aus diesen Erkenntnissen lassen sich wesentliche Anforderungen für die Gestaltung entwurfsunterstützender Werkzeuge ableiten.

4.4.1. FLEXIBILITÄT DES DENKENS

Routinen des Denkens Häufig verhindern festgefahrene Situationen (das sprichwörtliche „Brett vor dem Kopf“), unbewusste Einstellungen oder ungerechtfertigte Annahmen das Entdecken eines neuen Lösungsmusters [Banyard u. a., 1995; Pretz u. a., 2003]. Fehleinstellungen bezüglich des Problems, die das in Betracht gezogene Operatorinventar ungerechtfertigt einschränken, wurden bereits in Abschnitt 4.3.2 erwähnt.

Flexible Repräsentationen Eine Überwindung festgefahrener Situationen und damit auch das Entstehen spontaner Einfälle sind eng mit einer flexiblen gedanklichen Repräsentation der Problemsituation verknüpft. So gab Norman R. F. Maier seinen Studenten folgende Hinweise:

„Halten Sie Ihren Sinn offen für neue Bedeutungen. Das Lösungsmuster erscheint plötzlich. Sie können es nicht erzwingen. Bleiben Sie aufgeschlossen für neue Kombinationen und verschwenden Sie keine Zeit für erfolglose Bemühungen.“ ([Maier, 1933] nach [Krech u. a., 1992, S. 129])

Inkubationszeit Dörner beschreibt einen Zusammenhang zwischen einer (analysierenden) Phase intensiver gedanklicher Auseinandersetzung mit dem Problem, einer anschließenden, als *Inkubationszeit* bezeichneten Phase der Erholung und unbewussten geistigen Arbeit sowie dem verstärkten Entstehen spontaner Einfälle während dieser Erholungsphase [Dörner, 1976]. Er sieht in diesem Zusammenhang zwei gemeinsam wirkende Faktoren: Auf der einen Seite muss eine anfängliche intensive bewusste Auseinandersetzung die gedank-

liche Arbeit gewissermaßen in Gang bringen. Anschließend jedoch wirkt eine eher geringe Disziplin des Gedankenganges begünstigend, da dann die Verknüpfung gedanklicher Inhalte durch Assoziationsverläufe weniger gehemmt wird und so neue Einfälle wahrscheinlicher werden. Diese eher geringe Disziplin des Gedankenganges findet sich in der Regel in Phasen relativer Entspannung, in denen keine bewussten gedanklichen Anstrengungen unternommen werden.

*Wenig disziplinierter
Gedankengang*

Krech u. a. beschreiben verschiedene gezielt anwendbare Strategien zur Überwindung festgefahrener Situationen [Krech u. a., 1992], die sich im Wesentlichen mit den Erläuterungen von Dörner decken:

Handlungsstrategien

- Das bewusste **Vermindern der Aufmerksamkeit**, die man dem zu lösenden Problem widmet, kann dazu beitragen, übertriebene Konzentration und Fokussierung auf bestimmte Aspekte zu lösen.
- Durch **Umstrukturieren** des Problems und bewusstes **Ausprobieren** verschiedener Konstellationen erhöht sich die Wahrscheinlichkeit, zufällig eine Konstellation zu finden, in der ein Lösungsmuster erkennbar wird oder zumindest leichter gefunden werden kann.
- Das **Denken vom Ziel aus** und damit das Umkehren des Lösungsweges kann ausweglose oder festgefahrene Situationen umgehen.
- Durch **Abstrahieren** des Problems und die Suche nach **Analogien** können unter Umständen Lösungen aus anderen Bereichen gefunden und übertragen werden (vgl. auch S. 83).

Brodbeck verweist darauf, dass in erster Linie Unachtsamkeit sowie bewusstes oder unbewusstes routiniertes Verhalten Kreativität hemmen [Brodbeck, 1995]. Eine Herausforderung für den Entwerfenden besteht demnach darin, diese Routinen zu beobachten, zu erkennen und zu verändern. In diesem Zusammenhang betont Brodbeck die Bedeutung von Achtsamkeit in Form offener, entspannter Aufmerksamkeit gegenüber allen Aspekten der

Achtsamkeit

Problemsituation sowie der Situation in ihrer Gesamtheit. Denn auf diese Weise wird das Entdecken neuer Bedeutungen wahrscheinlicher.

4.4.2. FORMALE UND INFORMALE LÖSUNGSSTRATEGIEN

Bevorzugung informaler Vorgehensweisen Tendenziell meiden Menschen eher die Anwendung formaler symbolischer Lösungsverfahren und bevorzugen stattdessen kognitiv weniger anspruchsvolle, dafür aber hochinteraktive Handlungsweisen, die Aktivität auf verschiedenen Ebenen erfordern (Motorik, Wahrnehmung, Denken).

Interaktion und Rückkopplung Zimmermann und Campillo illustrieren diese Aussage am Beispiel des Autofahrens [Zimmermann u. Campillo, 2003]. Verschiedene Situationen im Straßenverkehr erfordern vom Autofahrer das Abbremsen seines Fahrzeugs. Die Aufgabe besteht dabei u. a. in der Bestimmung der erforderlichen Krafteinwirkung auf das Bremspedal. Die Lösung ist von einer Vielzahl von Faktoren abhängig, insbesondere der Masse und Geschwindigkeit des Fahrzeugs, der Qualität der Reifen, der Abnutzung der Bremsbeläge und der Beschaffenheit der Straßenoberfläche. Ein Autofahrer löst das Problem nicht durch eine formale Berechnung, sondern steuert die auf das Bremspedal ausgeübte Kraft anhand der unmittelbar wahrgenommenen Reaktion des Fahrzeugs.

Effiziente interaktive Lösungsfindung Für den Autofahrer ist das Problem des Bremsens in diesem Moment ein nicht exakt bestimmbares (*schlecht strukturiertes/ill-defined*) Problem. In einer normalen Straßenverkehrssituation ist keine Zeit zur Erfassung und formalen Repräsentation des Problems und zur Anwendung eines formalen Lösungsverfahrens. Aufgrund der direkten Interaktion mit dem Fahrzeug und der dadurch gegebenen Rückkopplung zu den Aktionen des Fahrers wird jedoch eine effiziente Lösung durch „Probieren“ möglich.

Expertise und systematische Vorgehensweisen Aber auch für zeitlich weniger kritische Probleme wie beispielsweise das Eingrenzen, Finden und Beheben von Fehlfunktionen oder Defekten an

Fahrzeugen weisen Zimmermann und Campillo auf den bevorzugten Einsatz probierender Handlungsstrategien hin. Der Einsatz formaler systematischer Lösungsmethoden nimmt meist mit dem Grad der Expertise einer Person auf einem bestimmten Gebiet zu. Voraussetzung dafür ist jedoch die Entwicklung grundlegender Kenntnisse des betreffenden Gebietes, seiner typischen Strukturen sowie Problem-Lösungs-Konstellationen [Zimmermann u. Campillo, 2003; Marchionini, 1997].

4.4.3. WIEDERVERWENDUNG VON ENTWURFSWISSEN

Der in Abschnitt 4.3 erläuterte Iterationsprozess und die zugehörigen Entwicklungsrichtungen Konkretisierung und Externalisierung beschreiben zunächst nur die verschiedenen Aktivitäten in Entwurfsvorgängen, deren Abfolge und deren Ergebnisse in Form mentaler oder externalisierter Repräsentationen des Entwurfsgegenstandes. Ein wesentlicher Aspekt des Entwurfs bleibt dabei jedoch unberücksichtigt: Die Frage, *warum* diese oder jene Entwurfsentscheidung getroffen wird; warum eine Lösung in einer bestimmten Art und Weise entwickelt und nicht eine andere der vielleicht in Frage kommenden Alternativen gewählt wurde.

Was sind die Gründe für eine Entwurfsentscheidung?

Entwurfsentscheidungen müssen stets ein Geflecht sowohl rationaler, objektiv messbarer Kriterien als auch nicht objektiv erfassbarer Wertvorstellungen und künstlerischer Ansprüche berücksichtigen. Oft sind dabei entgegengesetzt wirkende Kräfte auszugleichen. Typisch ist das wiederholte Auftreten von ähnlichen Problemkonstellationen.

Problemkonstellationen

Aufgrund von Erfahrungen entwickeln Architekten persönliches Entwurfswissen in Form generischer Lösungsprinzipien, die für bestimmte Problemkonstellationen erfolgreich angewendet werden können. Entwurfsentscheidungen werden so auf der Basis von (persönlichen) Erfahrungen mit bereits erfolgreich angewendeten Lösungen begründbar. Durch eine Dokumentation

Persönliches Entwurfswissen

solcher Problem-Lösungs-Paare kann Entwurfswissen interpersonell verfügbar und für das Finden begründbarer Entwurfslösungen nutzbar gemacht werden.

*C. Alexanders
Mustersprache* Den wohl bedeutsamsten Beitrag zur Dokumentation erfolgreicher bzw. erprobter Entwurfslösungen stellt die in den späten 70er Jahren von Christopher Alexander u. a. veröffentlichte Mustersprache dar. (Dt. Ausgabe: [Alexander u. a., 1995]) Die Autoren gehen dabei von der Annahme aus, dass Entwurfslösungen immer aus einem gegebenen Kontext, einer Problemstellung heraus entstehen und dass sich eine Reihe von Problemklassen identifizieren lassen, die universellen Charakter haben. Das heißt, dass es eine Reihe von Problemen gibt, die beim Entwerfen von Städten, Gebäuden und Konstruktionsdetails immer wieder auftreten. Ein weiterer Ausgangspunkt der Arbeiten ist die Feststellung, dass alle für eine bestimmte Problemklasse gefundenen Lösungen mehrere Merkmale gemeinsam haben. Anhand empirischer Untersuchungen entwickelten die Autoren einen Katalog von 253 untereinander in Beziehung stehenden Mustern.

„Jedes Muster beschreibt zunächst ein in unserer Umwelt immer wiederkehrendes Problem, beschreibt dann den Kern der Lösung des Problems, und zwar so, dass man diese Lösung millionenfach anwenden kann, ohne sich je zu wiederholen ” [Alexander u. a., 1995, X - Gebrauchshinweise]

*Muster als
externalisiertes
Expertenwissen* Die Muster stellen damit eine Form externalisierten Expertenwissens für den Problembereich des Architekturentwurfs dar. Alexander u. a. analysierten jeweils eine Reihe von Lösungen für ein ihnen bekanntes Problem und stellten die gefundenen übereinstimmenden Merkmale aller analysierten Lösungen als abstrakten Kern der Lösung in jeder Musterbeschreibung zusammen [Alexander u. a., 1995]. Die Muster dokumentieren so die Essenz von aus Erfahrung gewonnenem Entwurfswissen, auf einem Niveau, das die Wiederverwendung des Wissens für ähnliche Problemstellungen ermöglicht.

Im Bereich Architektur und Stadtplanung ist jedoch außer [Alexander u. a., 1995] keine weitere Quelle bekannt, die den Musteransatz thematisiert. In anderen Disziplinen dagegen konnte sich der Musteransatz sehr gut etablieren. So griffen unter anderem Arbeitsgruppen aus dem Bereich der objektorientierten Softwareentwicklung den Ansatz auf und veröffentlichten Mitte der 90er Jahre erste Musterkataloge, die Entwurfswissen aus der Entwicklung objektorientierter Softwaresysteme dokumentieren. Dazu siehe z. B. [Gamma u. a., 2004; Buschmann u. a., 1998]. Auf diese Arbeiten kommt Abschnitt 8.2.2 noch zurück.

*Resonanz des
Musteransatzes*

4.5. ENTWICKLUNG MENTALER MODELLE ALS HAUPTAUFGABE DES ENTWERFENS

Die Entwicklung mentaler Repräsentationen der Entwurfslösung muss als die hauptsächliche Aufgabe beim Entwerfen betrachtet werden. Ohne das Vorhandensein entsprechender mentaler Modelle ist eine Externalisierung in Form von Entwurfsdokumentationen nicht möglich. Externalisierte Repräsentationen dienen dabei als Arbeitsmittel während des Entwerfens. Mit ihrer Hilfe werden Lösungsideen festgehalten. So ist ein gedankliches Überprüfen der Lösung aber auch eine spätere Reinterpretation des Festgehaltenen möglich. Dabei bestimmt der Grad der Unschärfe, den eine externalisierte Repräsentation enthält, das Potential für eine spätere Reinterpretation [Gero, 1999]. Dies bedeutet, die eigentliche treibende Kraft des Entwurfsprozesses ist die (gedankliche) Arbeit an mentalen Repräsentationen, die jedoch in Wechselwirkung mit externalisierten Repräsentationen zu sehen ist, wie Abbildung 4.7 zeigt. Durch die Bewertung und Reinterpretation externalisierter Repräsentationen kann dabei wiederum eine Modifikation und Weiterentwicklung mentaler Repräsentationen erfolgen.

*Arbeit mit mentalen
Modellen*

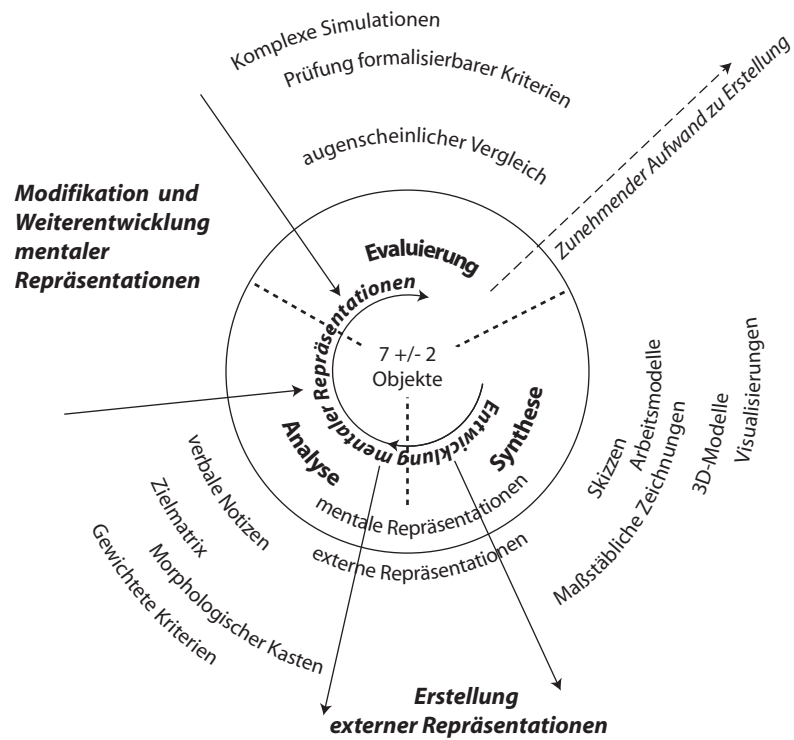


Abbildung 4.7.: Wechselwirkung mentaler und externalisierter Repräsentationen

Die Abbildung ist als eine „Draufsicht“ auf die von Asimov beschriebene Entwurfsspirale zu verstehen, wobei verschiedene Arten mentaler und externalisierter Repräsentationen bzw. darauf ausgeführter Operationen (Evaluation) eingetragen sind. Hier wird eine wichtige Abhängigkeit zwischen mentalen und externalisierten Repräsentationen deutlich: Nach [Dörner, 1976] erfordert jedes Ausführen einer Operation auf mentalen oder externalisierten Repräsentationen einen bestimmten kognitiven und zeitlichen (ggf. auch motorischen und materiellen) Aufwand. Für Operationen auf mentalen Modellen ist dieser in der Regel geringer als für Operationen auf externalisierten Modellen. Hinzu kommt, dass gedanklich ausgeführte Operationen jederzeit mit geringem Aufwand rückgängig gemacht werden können, was für Operationen auf externalisierten Modellen nicht oder nur mit deutlich höherem Aufwand möglich ist. Andererseits ist die Kapazität des Kurzzeitgedächtnisses begrenzt auf etwa $7 (\pm 2)$ Einheiten [Dörner, 1976]. Dies bedeutet sehr vereinfacht formuliert, je feingranularer die Repräsentationen werden und je mehr einzelne Komponenten sie enthalten, umso mehr ist ein Entwerfender auf die „Unterstützung“ des Kurzzeitgedächtnisses durch externalisierte Repräsentationen angewiesen. Mit wachsender Granularität der externalisierten Repräsentationen wächst aber auch der für ihre Erstellung und Bearbeitung erforderliche Aufwand. Für eine möglichst hohe Flexibilität der Repräsentationen ist es daher nicht sinnvoll, wenn die im Entwurf verwendeten Softwarewerkzeuge eine Externalisierung jedes einzelnen gedanklichen Schrittes notwendig machen.

*Abhängigkeit
zwischen mentalen
und externalisierten
Repräsentationen*

Softwarewerkzeuge können nur die Erstellung und Bearbeitung externalisierter Repräsentationen unterstützen. Genau an diesem Punkt ist ein entscheidendes Potential computerbasierter Werkzeuge zu sehen: in der Unterstützung bei der Organisation und beim Management externalisierter Repräsentationen, sobald diese ohnehin angefertigt werden. Dabei darf jedoch kein Zwang zu einer verfrühten Externalisierung oder Konkretisierung

*Externalisierung
nicht verfrüht
erzwingen*

entstehen, da auf diese Weise die flexible Weiterentwicklung mentaler Repräsentationen behindert wird. Ein in den Grundzügen ähnlicher Ansatz wurde u. a. bereits in [Steinmann, 1997] vertreten.

4.6. KONSEQUENZEN FÜR ENTWURFSUNTERSTÜTZENDE WERKZEUGE

Beantwortung des Fragekomplexes b) Die eingangs dieses Kapitels gestellten Fragen des Komplexes b) zum Entwerfen können nun folgendermaßen beantwortet werden:

Das Lösen von Entwurfsaufgaben erfolgt grundsätzlich in einem iterativen Prozess. Dabei stehen die Bildung und Weiterentwicklung mentaler Repräsentationen der Entwurfslösung im Mittelpunkt. Das Vorhandensein mentaler Repräsentationen ist die Voraussetzung für die Erstellung externalisierter Repräsentationen.

Softwaresysteme können nur die Erstellung und Bearbeitung externalisierter Repräsentationen unterstützen oder teilweise übernehmen. Die Verantwortung für die Entwicklung der mentalen Repräsentationen liegt vollständig beim Entwerfenden.

Eine wesentliche Voraussetzung für die Entwicklung mentaler Repräsentationen einer Entwurfslösung ist die Flexibilität des Denkens.

Anforderungen an entwurfsunterstützende Werkzeuge Softwaresysteme können und sollten diese Entwicklung vor allem dadurch unterstützen, dass sie die gedankliche Flexibilität des Entwerfenden so wenig wie möglich einschränken. In diesem Zusammenhang sind folgende Aspekte von Bedeutung:

- Unterstützung informaler Vorgehensweisen auf der Basis bewussten Probierens durch unmittelbare Interaktion und direkte Rückkopplung

- Keine oder wenig Bindung bzw. Kanalisation kognitiver Ressourcen¹ für die Ausführung sekundärer Tätigkeiten (z. B. Manipulation eines Computersystems und Interpretation von Ausgaben)
- Orientierung auf Anschaulichkeit und zugleich Zulassen von Redundanz und Unschärfe in externalisierten Repräsentationen.

Im Entwurfsprozess sind aus der Perspektive der iterativen Vorgehensweise drei verschiedene Ebenen komplexer, offener, nicht vollständig bestimmter Probleme zu lösen: Informationssuche, Lösungsfindung und Entscheidungsfindung. Der Zugriff auf eine digitale Bauwerksdokumentation wird dabei primär aus dem Kontext von Informationsproblemen erfolgen.

Die Informationsbeschaffung während des Entwerfens muss als ein integraler Bestandteil eines offenen, insgesamt nicht vorhersehbaren Prozesses betrachtet werden. Informationssuche wird parallel zu gedanklichen Prozessen der Lösungs- und Entscheidungsfindung ausgeführt. Dies bedeutet, dass für Werkzeuge zur Informationsbeschaffung die selben Ansprüche gelten, wie sie oben für entwurfsunterstützende Werkzeuge allgemeinen formuliert wurden.

Im nächsten Schritt sollen nun konkrete Anforderungen und Randbedingungen der Informationsbeschaffung herausgearbeitet werden. Vor diesem Hintergrund widmet sich das folgende Kapitel einer konkreten Betrachtung von Vorgehensweisen bei der Informationssuche.

¹ Gedächtnisleistungen, die u. a. zur Wahrnehmung, zur Steuerung der Aufmerksamkeit, des Denkens und der Informationsverarbeitung erforderlich sind

4. ENTWERFEN ALS KREATIVES PROBLEMLÖSEN

5. INFORMATIONSSUCHE

Im vorangegangenen Kapitel wurden ausgehend von der Betrachtung des Entwerfens als eine Verstandesleistung allgemeine Anforderungen an entwurfsunterstützende Werkzeuge formuliert. Diese sind zunächst als kategorisch zu betrachten und erfordern eine Präzisierung für den Kontext der entwurfsbezogenen Informationssuche. Vor diesem Hintergrund steht nun die Beantwortung des in Kapitel 1 formulierten Fragekomplexes c) (siehe S. 12) im Mittelpunkt: Wie gehen Menschen bei der Suche nach Informationen generell vor? Wodurch wird der Erfolg der Suche, vor allem in unbekanntem Informationsumgebungen, beeinflusst?

Ausgangspunkt

Gegenstand dieses Kapitels sind in erster Linie Vorgehensweisen und Ansprüche seitens der Informationssuchenden. Darüber hinaus wird die Beschaffenheit mentaler Repräsentationen hinterfragt, da diese eine wesentliche Steuergröße der (entwurfsbezogenen) Informationssuche darstellen. Aus diesen beiden Aspekten werden Kriterien zur Gestaltung der zu entwickelnden Informationsumgebung abgeleitet.

Inhalt

5.1. ABGRENZUNG

Informationssuche wird im Rahmen dieser Arbeit aus der Perspektive eines Suchenden betrachtet, nicht vom Standpunkt eines die Suche unterstützenden Systems. Das bedeutet, alle beschriebenen Aktivitäten, Vorgehensweisen und Ergebnisse sind als Handlungen bzw. Ziele von Personen zu verste-

Standpunkt und Perspektive

hen, die eine Informationsumgebung verwenden – nicht als Eigenschaften der Informationsumgebung.

Kognitiver Prozess Informationssuche soll als ein offener Prozess verstanden werden, den Menschen gezielt in Gang setzen, um ihren persönlichen Wissensstand zu erweitern [Marchionini, 1997; Taube, 1998]. Offenheit bedeutet, es muss nicht unbedingt ein Ergebnis gefunden werden. Ein Informationssuchender kann die Suche auch ohne Erfolg abbrechen. **Information** steht in diesem Rahmen für all das, was das Potential hat, das Wissen einer Person zu verändern. Aus dieser Perspektive ist Informationssuche als ein fundamentaler kognitiver Prozess zu sehen, der oft Teil des Lernens oder des Problemlösens ist – nicht primär als Aufgabe oder Funktionalität eines elektronischen Systems. Im Kontext dieser Arbeit soll Informationssuche konkret als ein Bestandteil der Verstandesleistung Entwerfen betrachtet werden.

Unterschied zum Information Retrieval Diese Auffassung der Informationssuche unterscheidet sich grundsätzlich von der des *Information Retrieval*, das sich auf die Ermittlung aller für ein bereits exakt definiertes Suchziel relevanten Objekte aus einer Datenbank sowie die dafür benötigten Algorithmen und Datenstrukturen konzentriert [Colomb, 2002]. Die in dieser Arbeit zugrunde gelegte Sichtweise der Informationssuche schließt explizit die Person des Suchenden mit ein und betrachtet den Suchenden, dessen Informationsproblem sowie die Informationsumgebung als Teile eines Gesamtsystems [Vester, 1999], die jeweils im Zusammenspiel Erfolg oder Misserfolg der Suche bedingen. Bestandteil der Informationssuche sind somit auch alle Arten kognitiver und physischer Aktivitäten des Suchenden bei der Erfassung, dem Verstehen und Beschreiben sowie ggf. Formalisieren des Informationsproblems, aber auch Strategien und Vorgehensweisen, deren sich ein Informationssuchender bedient.

Wechselwirkung zwischen Informationsumgebung und Informationssuche Informationssuche findet stets als Interaktion zwischen einem Informationssuchenden und einer Informationsumgebung statt. Dabei beeinflusst die Struktur und Beschaffenheit der Informationsumgebung die Vorgehens- und

Verhaltensweisen des Suchenden, indem sie natürliche Verhaltensweisen unterstützt und/oder das Erlernen neuer Fähigkeiten erfordert. Umgekehrt ist jedoch auch eine Einflussnahme möglich, indem die Ansprüche der Informationssuchenden den Ausgangspunkt für die Gestaltung von Informationsumgebungen bilden, die ihrerseits wiederum Ansprüche an die Fähigkeiten des Suchenden stellen bzw. Erwartungshaltungen entstehen lassen. Beides kann daher nicht als voneinander getrennt betrachtet werden.

5.2. BEGRIFFE

Die folgenden Abschnitte dienen zunächst der inhaltlichen Bestimmung einiger für den Diskursbereich der Informationssuche zentralen Begriffe.

5.2.1. INFORMATIONSPRAUM UND NAVIGATION

Die Begriffe Informationsraum und Navigation stellen zwei eng miteinander verknüpfte Konzepte dar, die als Metaphern für den Umgang mit umfangreichen Informationsbeständen, insbesondere bei der Suche nach Informationen dienen.

Metaphern

In Anlehnung an [Colomb, 2002; Marchionini, 1997; Taube, 1998] soll der Begriff des **Informationsraums** in dieser Arbeit als Metapher für eine strukturierte Sammlung von Informationseinheiten dienen, in der sich Menschen zwischen einzelnen Informationseinheiten bewegen können – ähnlich der Bewegung in physischen (geografischen, urbanen oder architektonischen) Räumen. Dies setzt einerseits voraus, dass einzelne Informationseinheiten für Menschen identifizierbar sind. Andererseits müssen die Informationseinheiten untereinander ein strukturelles Gefüge aufweisen, das Menschen ähnlich wie ein Wegenetz zur Bewegung zwischen den Informati-

*Strukturierte Menge
von Informationseinheiten*

onseinheiten benutzen können. Die Repräsentationsform des Informationsraums (physisch oder digital/rechnerintern) spielt für die Anwendung der Metapher zunächst keine Rolle.

Informationsraum *Def. Informationsraum:* Unter einem Informationsraum wird eine strukturierte Menge von Informationseinheiten verstanden, die physisch oder digital repräsentiert sein kann. Die durch die Beziehungen zwischen den Informationseinheiten gebildeten Strukturen können dabei von Menschen zur Bewegung durch den Informationsraum verwendet werden.

Orientierung in physischen Räumen Die Metapher des Informationsraums erlaubt eine metaphorische Übertragung von Mechanismen zur Orientierung in physischen Räumen auf die Orientierung und Bewegung im Informationsraum. Eines der wichtigsten Konzepte in diesem Zusammenhang ist das der Navigation.

Der Begriff **Navigation** stammt ursprünglich aus der Raum- bzw. Schifffahrt und steht dort für das Bestimmen des aktuellen Standpunktes und das Festlegen des Kurses, um ein angestrebtes Ziel zu erreichen [Drodowski, 1991; Marchionini, 1997].

Menschliche Aktivität Im Rahmen dieser Arbeit soll Navigation als eine aktive Tätigkeit menschlicher Individuen verstanden werden, die einerseits die Orientierung in einer Umgebung und andererseits die Koordination und Steuerung der eigenen Bewegung durch die Umgebung beinhaltet. Dabei strebt das Individuum ein bestimmtes Ziel an.

Navigation bedeutet für eine Person unter anderem die Beantwortung der folgenden Fragen:

- Wo bin ich?
- Wie bin ich hierher gekommen?/Wie komme ich zurück?
- Was kann ich hier tun?

- Wohin kann ich von hier aus gelangen?

Diese Auffassung von Navigation kann sehr gut als Metapher zum Umgang mit digitalen Umgebungen, insbesondere digital repräsentierten Informationsräumen dienen. Siehe dazu [Colomb, 2002; van Steenis, 1992; Kubicek, 1997; Kuhlen, 1997; Marchionini, 1997]. Innerhalb dieser Arbeit wird der Begriff Navigation somit folgendermaßen verwendet:

Def. Navigation: Navigation bezeichnet die Orientierung eines Individuums in einer realen oder digital repräsentierten Umgebung, wobei das Individuum seine Bewegung durch diese Umgebung aktiv steuert, um ein bestimmtes Ziel zu erreichen. *Navigation*

5.2.2. DIREKTE MANIPULATION

Direkte Manipulation ist ein wesentliches Konzept interaktiver Nutzerschnittstellen und spielt im Rahmen der Arbeit eine wichtige Rolle bei der interaktiven Suche in digital repräsentierten Informationsräumen.

*Interaktion mit
Informationsräumen*

Diese Form der Mensch-Maschine-Interaktion baut auf eine grafische Nutzerschnittstelle mit folgenden Eigenschaften [Stary, 1996]:

- Bestimmte Objekte werden permanent auf dem Bildschirm zur Manipulation vorgehalten.
- Diese Objekte können durch physikalische Gesten und die direkte Auswahl und Ausführung von Funktionen manipuliert werden. Benutzer sind nicht gezwungen, Kommandos über die Tastatur einzugeben bzw. die Grammatik von Kommandosprachen zu memorieren.
- Die auf den angezeigten Objekten ausgeführten Operationen sind unmittelbar sichtbar (Feedback) und reversibel.

5.2.3. INFORMATIONSUMGEBUNG

*Informationsumgebung
bzw. Suchsystem* Marchionini versteht unter einer **Informationsumgebung** allgemein alle Einrichtungen, von denen Menschen Informationen beziehen können [Marchionini, 1997]. Dies können sowohl soziale Netzwerke, Freunde oder Kollegen, Bibliotheken oder einzelne Bücher, aber auch das Internet, einzelne Websites oder spezielle Datenbanken sein. Synonym für Informationsumgebung verwendet Marchionini auch die Bezeichnung *Suchsystem*.

*Informationsraum
bzw. Datenbasis* Ferner unterscheidet Marchionini Datenbasis und Schnittstelle als zwei Bestandteile eines Suchsystems. Die **Datenbasis** umfasst das potentiell durch das Suchsystem verfügbare Wissen. Zu den Merkmalen der Datenbasis zählen dabei nicht nur die Menge der verfügbaren Informationseinheiten und deren inhaltliche Beziehungen, sondern auch das Medium, das der Repräsentation der Informationseinheiten dient und dessen physische Organisation. In Orientierung an [Taube, 1998; Colomb, 2002] wird im Folgenden der Begriff *Informationsraum* synonym zu Datenbasis verwendet. Darunter soll allgemein eine strukturierte Menge von Informationseinheiten verstanden werden.

Schnittstelle Unter der **Schnittstelle** eines Suchsystems versteht Marchionini die Gesamtheit aller Repräsentationen, Werkzeuge, Regeln und Mechanismen, die Informationssuchenden den Zugriff auf den Informationsraum und dessen Manipulation ermöglichen. Bei elektronischen Systemen wird meist explizit die Bezeichnung *Nutzerschnittstelle* verwendet.

Abbildung 5.1 verdeutlicht den Zusammenhang zwischen Suchsystem bzw. Informationsumgebung, Informationsraum bzw. Datenbasis und Schnittstelle.

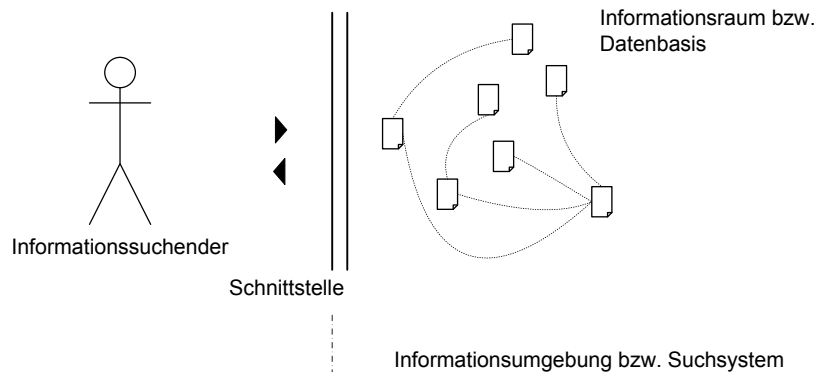


Abbildung 5.1.: Informationsumgebung, Informationsraum und Schnittstelle

5.2.4. INFORMATIONSPROBLEM

Unter Bezugnahme auf die allgemeine Definition eines Problems (vgl. S. 73) und in Anlehnung an [Marchionini, 1997] kann ein **Informationsproblem** als ein Zustand der Unwissenheit oder Unsicherheit bezüglich eines Sachverhaltes verstanden werden, den eine Person in einen Zustand des Wissens überführen möchte. Die Barriere besteht in diesem Fall in Form eines Informationsdefizits, wobei die drei von Dörner genannten Barriertypen auftreten können (vgl. [Dörner, 1976] sowie Seite 80 dieser Arbeit). Das heißt, Informationsprobleme können anhand der Klarheit bzw. des Formalisierungsgrades des Suchziels sowie anhand der verfügbaren Kenntnisse bezüglich des Informationsraums und des Suchsystems klassifiziert werden in Interpolations-, Synthese- und dialektische Probleme [Taube, 1998]. Dabei kann ebenfalls eine Überlagerung verschiedener Barrieren auftreten.

*Zustand der
Unwissenheit oder
Unsicherheit*

5.3. INFORMATIONSSUCHE ALS PROBLEMLÖSEPROZESS

Die folgenden Abschnitte betrachten kognitive Aktivitäten und Vorgehensweisen der Informationssuche sowie deren Wechselwirkung mit verschiedenen Barrieretypen bei Informationsproblemen.

5.3.1. TEILPROZESSE DER INFORMATIONSSUCHE

Marchionini identifiziert acht Teilprozesse der Informationssuche, die er in drei Klassen zusammenfasst (siehe [Marchionini, 1997] sowie Tabelle 5.1). Die drei Klassen entsprechen den im vorangegangenen Kapitel beschriebenen Phasen *Analyse*, *Synthese* und *Evaluierung*.

Prozessklasse	Teilprozess
<i>Verstehen des Problems [Analyse]</i>	1) Erkennen und Akzeptieren des Informationsproblems 2) Definieren und Verstehen des Informationsproblems
<i>Planen und Ausführen der Suche [Synthese]</i>	3) Wahl der Informationsquelle(n) bzw. des Suchsystems 4) Formulieren einer Anfrage 5) Ausführen der Suche
<i>Bewertung und Verwendung der (Zwischen-)Ergebnisse [Evaluierung]</i>	6) Untersuchung der Ergebnisse 7) Extraktion von Informationen (aus dem Ergebnis) 8) Reflektion / Iteration / Abbruch (Entscheidung über weitere Schritte)

Tabelle 5.1.: Teilprozesse der Informationssuche nach [Marchionini, 1997]

Diese Teilprozesse sind nicht als sequenziell ablaufende Schritte zu verstehen, sondern als *Aktivitätsmodule*, die parallel oder nacheinander ablaufen können. Sie bleiben teilweise über den gesamten Verlauf der Suche aktiv, verwenden sich gegenseitig oder rufen sich rekursiv auf (vgl. Abbildung 5.2).

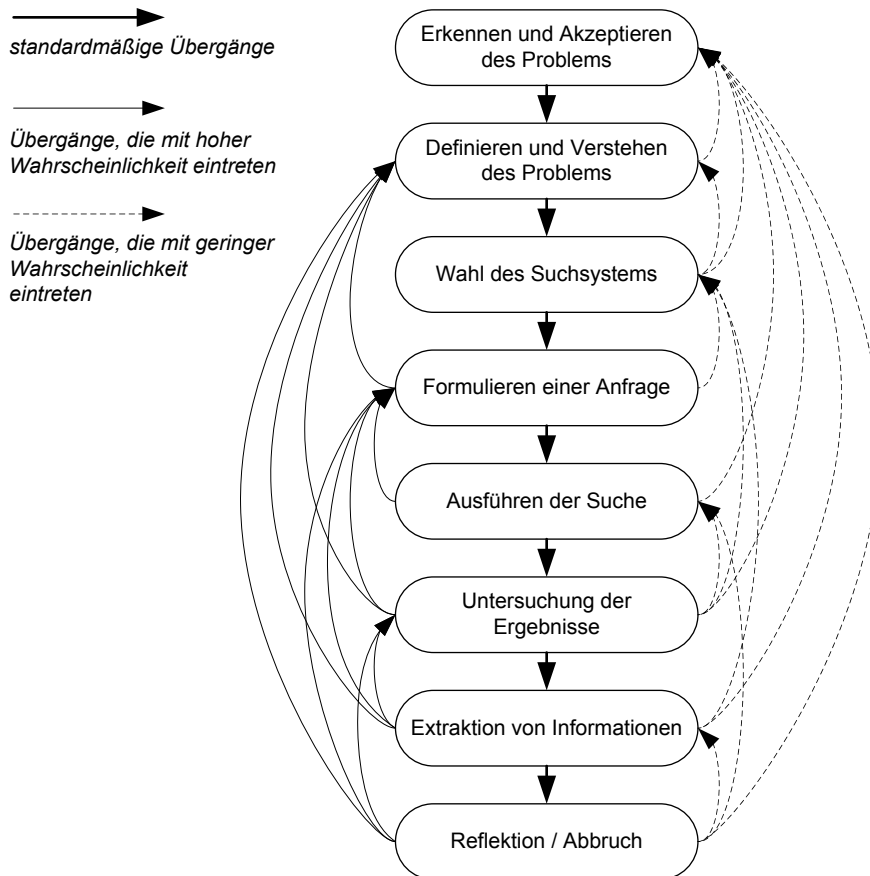


Abbildung 5.2.: Übergänge zwischen den Teilprozessen der Informationssuche nach [Marchionini, 1997, S. 50]

Das Erkennen und Akzeptieren des Informationsproblems initiiert den Suchprozess. Der nächste Schritt besteht im Aufbau einer mentalen Repräsentation des Informationsproblems. Das heißt, das Problem muss vom

Verstehen des Problems

Suchenden verstanden und definiert werden. Dieser Teilprozess bleibt in der Regel für den gesamten Verlauf der Suche aktiv.

Planen und Ausführen der Suche Mit der Wahl des Suchsystems und der Formulierung einer Anfrage muss die entwickelte mentale Repräsentation des Informationsproblems auf das Suchsystem abgebildet werden. Der Informationssuchende muss dabei seine Begriffswelt auf die vom System verwendete übertragen, sowie seine Vorstellungen hinsichtlich auszuführender Aktionen mit den vom Suchsystem unterstützten Aktionsmöglichkeiten abgleichen.

Bewertung und Verwendung der Ergebnisse Schließlich müssen die gefundenen Ergebnisse durch den Suchenden bewertet und ggf. verwendet werden. Dabei ist grundsätzlich zu berücksichtigen, dass digital repräsentierte Informationen nie direkt für Menschen zugänglich sind, sondern immer der Transformation in eine für Menschen interpretierbare Darstellung bedürfen. Die Präsentationsform, die das System zur Darstellung der gefundenen Informationsobjekte verwendet, beeinflusst somit maßgeblich die Möglichkeiten zur Untersuchung, Bewertung und Verwendung der Ergebnisse durch den Suchenden.

Mentale Repräsentationen als steuerndes Element der Suche Ausgangspunkt der Informationssuche sind stets mentale Repräsentationen des Suchziels, der notwendigen Aktionen der Suche sowie der verwendeten Informationsumgebung. Diese steuern sämtliche Aktivitäten und Interaktionen mit dem Suchsystem, die ein Suchender ausführt, wobei dieser permanent mentale Repräsentationen mit den wahrgenommenen Eigenschaften und Reaktionen des Suchsystems vergleicht. Parallel zum Verlauf der Suche werden diese mentalen Repräsentationen durch Reflektion weiter entwickelt, verfeinert und an die tatsächlich vorgefundenen Merkmale der Informationsumgebung bzw. des Suchsystems angepasst. Ein Suchsystem kann diese kognitiven Prozesse nur insofern unterstützen, als dass die zu überbrückenden Abweichungen zwischen bereits vorhandenen mentalen Repräsentationen bzw. Erwartungen seitens des Suchenden und den tatsächlichen Eigenschaften des Suchsystems auf ein Minimum reduziert wer-

den. Zwei wesentliche Einflussgrößen sind in diesem Zusammenhang:

- der Aufbau und die Strukturierung des Informationsraums sowie
- die Präsentationsformen und Interaktionsmechanismen der vom Suchsystem bereitgestellten Schnittstelle zum Informationsraum.

Wie aus den Aktivitätsmodulen ersichtlich ist, sind das Bilden und das Untersuchen von Teilmengen des Informationsraums elementare Aktivitäten der Suche, die die Schnittstelle zum Informationsraum unabhängig von spezifischen Präsentationsformen und Interaktionsmechanismen unterstützen muss.

*Bilden und
Untersuchen von
Teilmengen*

5.3.2. VORGEHENSWEISEN BEI DER SUCHE (SUCHSTRATEGIEN)

In der Literatur werden zwei Herangehensweisen von Personen an die Informationssuche beschrieben, die im Folgenden als Suchstrategien bezeichnet werden sollen. Sie unterscheiden sich nach der Art und Weise, wie Informationssuchende bei der Durchführung der Suche vorgehen, und entsprechen im Prinzip den in Abschnitt 4.4.2 allgemein erwähnten formalen und informalen Vorgehensweisen beim Problemlösen.

Bislang existiert kein Konsens bezüglich einer einheitlichen Verwendung der Bezeichnungen für die beiden Strategien. Im Rahmen dieser Arbeit sollen die Begriffe *analytische* und *explorative* Suche verwendet werden. Analytische Suche steht für die Suche mit Hilfe formaler Anfragen unter Verwendung einer strukturierten Anfragesprache. Explorative Suche steht für das interaktive Erkunden eines Informationsraumes, etwa durch Verfolgen von Hyperlinks. In englischsprachiger Literatur wird dafür die Bezeichnung *Browsing* verwendet. Vgl. dazu [Marchionini, 1997; Taube, 1998; Kuhlen, 1997; Colomb, 2002].

*Analytische und
explorative Suche*

*Skala praxisrelevanter
Strategien*

Analytische und explorative Suche in ihrer reinen Form können als Extreme einer Skala der in der Praxis angewendeten Suchstrategien verstanden werden. Welche Ausprägungen oder Kombinationen der Strategien Informationssuchende letztlich anwenden, hängt nicht zuletzt von den Möglichkeiten des Suchsystems und den vorhandenen Kenntnissen des Suchenden ab [Marchionini, 1997; Taube, 1998]. Beide Autoren plädieren daher für eine integrierte Unterstützung der verschiedenen Suchstrategien durch Informationssysteme. Denn auf diese Weise können Informationssuchende eine jeweils situationsbezogen angemessene Herangehensweise selbst wählen.

Tabelle 5.2 stellt die wesentlichen Merkmale analytischer und explorativer Suchstrategien gegenüber. Beide Strategien stellen unterschiedliche Anforderungen an den Suchenden aber auch an die Beschaffenheit der Informationsumgebung, diese finden sich in Tabelle 5.3.

*Randbedingungen
analytischer Suche*

Analytische Suche erfordert von den Suchenden überwiegend kognitive Aktivität in Form analytischen Denkens. Darüber hinaus ist die Kenntnis der vom Suchsystem unterstützten Anfragesprache notwendig. Parallel müssen die Suchenden die Strukturen und typischen Inhalte des Informationsraums sehr gut kennen, um sinnvolle Anfragen formulieren zu können. Zum Erwerb dieser Kenntnisse ist oft eine entsprechende Schulung oder zumindest Einarbeitung notwendig. Die Anwendung analytischer Suchstrategien erfordert folglich jeweils den Erwerb von Handlungswissen für eine konkrete Informationsumgebung. Dieses Handlungswissen stellt eine Spezialisierung auf die jeweilige Umgebung dar, für die es eine sehr effiziente Suche ermöglicht. Es kann jedoch nur eingeschränkt auf andere Informationsumgebungen übertragen werden.

*Randbedingungen
explorativer Suche*

Explorative Vorgehensweisen sind Bestandteil natürlicher menschlicher Verhaltensweisen des Lernens und des Erkundens einer Umgebung und müssen nicht von Grund auf neu erlernt werden. Sie basieren hauptsächlich auf Interaktivität und direkter Manipulation des Informationsraums. Dies

5.3. INFORMATIONSSUCHE ALS PROBLEMLÖSEPROZESS

<i>Charakteristische Merkmale</i>	Analytische Suche	Explorative Suche
<i>typischer Kontext bzw. Basis empirischer Untersuchungen</i>	Benutzung bibliografischer Online-Datenbanken durch professionelle Informationsvermittler	Lesen von Hypertext-/ Hypermedia-Dokumenten
<i>Kritische Faktoren des Kontextes</i>	Kostenminimierung durch Minimierung der Online-Nutzungszeit	(keine)
<i>Vorgehensweise</i>	Formaler Lösungsweg mit systematischer Planung im Vorfeld und anschließender Abarbeitung geplanter Anfragen	Heuristische, opportunistische Navigation durch den Informationsraum anhand situativ wahrgenommener Hinweise
<i>Grundlegendes Prinzip</i>	Spezifikation der gesuchten Objekte durch den Suchenden; Vergleich zwischen spezifizierten und tatsächlich existierenden Objekten durch das System (Matching-Paradigma)	Erkennen relevanter Objekte durch den Suchenden; Vergleich zwischen mentalen Repräsentationen und tatsächlich existierenden Objekten durch den Suchenden (Browsing-Paradigma)
<i>Eignung für Problemklassen</i>	Gut strukturierte Probleme / Interpolationsprobleme	Schlecht strukturierte Probleme / Synthese- und / oder dialektische Probleme
<i>Einordnung</i>	Spezialisierte Strategien für eine bestimmte Informationsumgebung, nur bedingt übertragbar	Nicht an eine Informationsumgebung gebunden, Effizienz wird aber wesentlich durch die Informationsumgebung bestimmt

Tabelle 5.2.: Merkmale analytischer und explorativer Suchstrategien

<i>Anforderungen</i>	Analytische Suche	Explorative Suche
<i>an die Fähigkeiten der Suchenden</i>	Erlernen einer strukturierten Anfragesprache, Einarbeiten in die Struktur des Informationsraums (hoher Lernaufwand)	Bestandteil des natürlichen Lernverhaltens (geringer Lernaufwand)
<i>an die Art der Aktivitäten während der Suche</i>	Überwiegend kognitive Aktivität; kanalisiertes, analytisches Denken	Kombination und fließender Wechsel von Wahrnehmung / Aufmerksamkeit, motorischer und kognitiver (reflektierender) Aktivität
<i>an die Organisation / Struktur des Informationsraums</i>	Systematische Erschließung, z.B. mit Hilfe einer Dokumentationssprache	Gute Strukturierung über mehrere Abstraktionsebenen, die durch die Schnittstelle für den Suchenden explizit gemacht wird
<i>an die Funktionalität des Suchsystems</i>	Bereitstellung einer Anfragesprache, Umsetzung des Matching-Paradigmas auf der Dokumentationssprache	Ermöglichen direkter Manipulation des Informationsraums und Interaktion mit Informationsobjekten

Tabelle 5.3.: Analytische und explorative Suchstrategien - resultierende Anforderungen

erfordert vom Suchenden eine permanent aktive Wahrnehmung und Aufmerksamkeit in Kombination mit motorischen und kognitiv-reflektierenden Aktivitäten. Durch diese Kombination mehrerer Aktivitätsebenen und den starken Einfluss von Motivation und Interessen auf den Suchvorgang werden explorative Suchstrategien als kognitiv weniger anspruchsvoll bewertet [Marchionini, 1997; Zimmermann u. Campillo, 2003]. Charakteristisch für explorative, d.h. informale Suchstrategien ist eine starke Verzahnung der einzelnen Teilprozesse (Aktivitätsmodule), so dass diese bei der Beobachtung oft nicht mehr klar voneinander unterschieden werden können.

Explorative Strategien sind für Suchende universeller einsetzbar, als formale analytische Vorgehensweisen. Elementare Voraussetzung ist jedoch, dass das Suchsystem die Bildung von Teilmengen des Informationsraums durch direkte Manipulation ermöglicht und unmittelbare Rückmeldungen auf einzelne Aktionen des Suchenden gibt. Vor allem bei elektronischen Informationsumgebungen besteht daher eine starke Abhängigkeit von den Möglichkeiten und Funktionalitäten des Systems.

*Notwendigkeit
direkter Manipulation
und guter
Strukturierung*

5.3.3. WECHSELWIRKUNGEN ZWISCHEN PROBLEMKLASSEN UND SUCHSTRATEGIEN

Die Anwendung explorativer Strategien kann Suchenden zur Vorbereitung stärker formalisierter bzw. systematischer Vorgehensweisen dienen, insbesondere dann, wenn die Klarheit des Suchziels bzw. Kenntnisse des Suchsystems zu Beginn der Suche nicht gegeben sind [Taube, 1998; Marchionini, 1997]. Durch experimentierendes Interagieren mit dem Suchsystem und spielerisches, interaktives Erkunden des Informationsraums können Suchende einerseits die mentalen Repräsentationen des Informationsproblems präzisieren und damit die Zielkriterien der Suche klären. Andererseits können mentale Modelle des Suchsystems und des Informationsraums entwickelt

*Klärung von
Zielkriterien und
Systemstruktur*

und angepasst werden, die wiederum der Planung weiterer Aktionen dienen. Abbildung 5.3 zeigt mögliche Überführungsrichtungen zwischen einzelnen Typen von Informationsproblemen durch Anwendung explorativer Suchstrategien (vgl. dazu auch Abbildung 4.3 auf S. 80 dieser Arbeit).

Bedeutung explorativer Suche und Interaktivität Dieser Zusammenhang unterstreicht noch einmal die Bedeutung explorativer Suchstrategien und direkter Interaktion mit dem Informationsraum, vor allem beim Umgang mit unscharfen Suchzielen und der Suche in nicht vertrauten Informationsumgebungen. Daraus resultiert die Forderung nach einer guten Unterstützung explorativer Suchstrategien beim Umgang mit einer lebenszyklusübergreifenden Bauwerksdokumentation und deren Verwendung in frühen Phasen der Planung im Bestand.

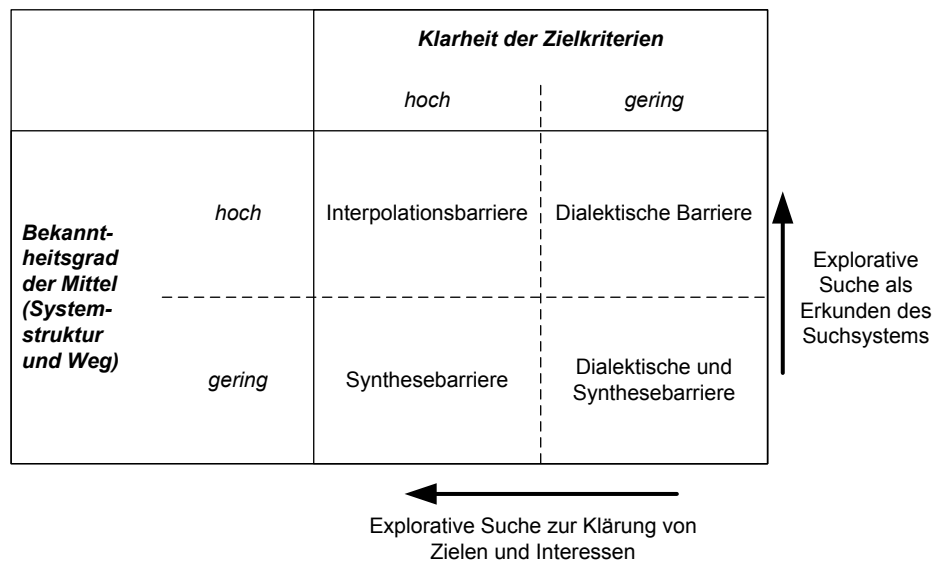


Abbildung 5.3.: Explorative Suche zur Klärung von Informationsproblemen nach [Taube, 1998, S. 158] auf Basis der Problemklassifikationsmatrix nach [Dörner, 1976]

5.4. ORGANISATION MENTALER REPRÄSENTATIONEN

Wie im vorangegangenen Abschnitt gezeigt wurde, ist der Vergleich zwischen mentalen Repräsentationen und den tatsächlich wahrnehmbaren Eigenschaften einer Informationsumgebung bzw. eines Informationsraums ein zentrales steuerndes Element der Informationssuche. Vor diesem Hintergrund untersuchen die folgenden Abschnitte Prinzipien der Organisation mentaler Repräsentationen. Dabei spielen insbesondere Repräsentationen, die Menschen bei der Orientierung in räumlichen Umgebungen verwenden, aber auch allgemeine Prinzipien zur Strukturierung bauwerksbezogener Informationen eine Rolle.

5.4.1. ORGANISATIONSPRINZIPIEN

Die Organisation von Inhalten im menschlichen Gedächtnis ist ein Untersuchungsgegenstand der Kognitionspsychologie. Diese beschreibt eine Reihe verschiedener Prinzipien der Informationsorganisation in mentalen Repräsentationen, die anhand empirischer Untersuchungen identifiziert wurden. Unter *Inhalten* werden in diesem Zusammenhang alle Arten gedanklicher Einheiten, wie Objekte, Eigenschaften, Tätigkeiten und Handlungsweisen etc. verstanden.

*Elementare
Organisations-
prinzipien*

Die Erkenntnisse der Kognitionspsychologie wurden u. a. in der *künstlichen Intelligenz* aufgegriffen und als Vorbilder zur formalen Repräsentation von (externalisiertem) Wissen verwendet. Diese Ansätze sollen hier jedoch nicht diskutiert werden; an dieser Stelle stehen ausdrücklich Aussagen zu mentalen Repräsentationen im Mittelpunkt.

*Formale
Repräsentationen von
Wissen*

Folgende Organisationsprinzipien sind bekannt:

- **Begriffe/Kategorien** fassen Inhalte mit ähnlichen Eigenschaften zusammen und dienen damit primär der Reduktion von Komplexität [Banyard u. a., 1995; Holzinger, 2001].
- **Assoziationen** verknüpfen als zusammengehörig empfundene Inhalte und erleichtern damit das Abrufen von Inhalten [Holzinger, 2001].
- **Semantische Netzwerke** sind Strukturen, die Begriffe und Assoziationen zur Repräsentation komplexer Zusammenhänge verwenden. Die Idee der semantischen Netzwerke bildete eine wesentliche Grundlage für verschiedene Ansätze der formalen externalisierten Repräsentation von Wissen in der künstlichen Intelligenz. Dazu siehe auch [Dörner, 1976; Banyard u. a., 1995; Holzinger, 2001].
- **Schemata** enthalten kategoriales Wissen einer Person zu einem bestimmten Sachverhalt, wie z.B. *Auto fahren*. Schemata sind dabei semantische Netzwerke, die Leerstellen aufweisen, in die bestimmte, einer konkreten Situation zugeordnete Inhalte eingefügt werden können, beispielsweise das *konkrete Fahrzeug* [Holzinger, 2001].
- **Skripte** als Spezialform von Schemata beschreiben in der Art eines Drehbuches Handlungsanweisungen und Erwartungen einer Person bezogen auf eine bestimmte Art von Situationen, wie beispielsweise den *Besuch eines Restaurants* [Holzinger, 2001].
- **Kognitive Landkarten** repräsentieren Merkmale konkreter räumlicher Umgebungen und dienen Menschen bei der Orientierung und Navigation in diesen Umgebungen [Banyard u. a., 1995; Lynch, 1965].

*Dynamik mentaler
Repräsentationen*

Die genannten Organisationsformen bauen aufeinander auf, wobei komplexere Formen jeweils die einfacheren als Bausteine verwenden. Mentale Repräsentationen müssen grundsätzlich als dynamisch verstanden werden.

Das heißt, sie entwickeln sich durch Erfahrungen und Lernprozesse weiter. Menschen sind darüber hinaus in der Lage, nur die Aspekte einer mentalen Repräsentation zu „aktivieren“, die für die augenblickliche Situation tatsächlich relevant sind. Nicht benötigte Aspekte werden dabei ausgeblendet. Dazu siehe auch S. 131 dieser Arbeit.

5.4.2. ORIENTIERUNG IM ENTWURFS- UND PLANUNGSGEGENSTAND BAUWERK

Kognitive Landkarten stehen in engem Bezug zur Orientierung innerhalb des Entwurfs- und Planungsgegenstandes Bauwerk bzw. Gebäude. Sie beziehen sich stets auf eine konkrete räumliche Situation, für die wahrgenommene räumlich-topologische Zusammenhänge im Gedächtnis gespeichert werden. Das Einprägen raumbezogener Merkzeichen ermöglicht menschlichen Individuen vor allem Bewegungsfreiraum in ihrer Umgebung und kann in Extremsituationen sogar überlebensnotwendig sein. Der Umgang mit kognitiven Karten wird daher prinzipiell als eine natürliche Verhaltensweise angesehen, über die Menschen verfügen, ohne sie explizit erlernen zu müssen. Dazu vgl. [Lynch, 1965].

*Orientierung in
räumlichen
Umgebungen*

Lynch untersuchte anhand von drei amerikanischen Städten, wie Menschen sich in urbanen Räumen orientieren und welche raumbezogenen Merkmale ihnen als Orientierungshilfen dienen. Daraus lassen sich wiederum Schlussfolgerungen zur Beschaffenheit der mentalen Modelle dieser urbanen Räume ziehen. In dieser Studie wurden drei wesentliche Merkmale räumlicher Objekte identifiziert, die den Aufbau mentaler Modelle beeinflussen:

*Relevante Merkmale
räumlicher Objekte*

- **Identität**

Ein räumliches Objekt besitzt für einen Beobachter Identität, wenn es als eine Ganzheit wahrgenommen werden kann, die aufgrund ihrer Individualität eindeutig identifizierbar ist.

- **Struktur**

Beziehungen zwischen räumlichen Objekten untereinander oder zum Beobachter bilden die Strukturen einer räumlichen Umgebung. Beziehungen ergeben sich durch räumliche Lage, Richtungen und Distanzen zwischen Objekten bzw. Objekten und Beobachter.

- **Bedeutung**

Bedeutung entsteht durch den praktischen oder gefühlsmäßigen Sinn, den ein räumliches Objekt für einen Beobachter besitzt. Verschiedene Beobachter können räumlichen Objekten unterschiedliche Bedeutungen zuordnen.

Orientierung in Bauwerken Bauwerke sind ebenfalls Teil der räumlichen Umwelt des Menschen. Somit kann davon ausgegangen werden, dass zur Orientierung in diesen ähnliche mentale Bilder der konkreten räumlichen Situation dienen wie bei der Orientierung in großmaßstäblichen urbanen Räumen.

Räumliche Kodierung planungsrelevanter Informationen Traditionell erfolgt die Kommunikation in Bauplanungsprojekten über Pläne oder dreidimensionale Modelle. Das bedeutet, das individuelle räumliche Gefüge eines Bauwerks dient als primäres Ordnungsprinzip für planungsrelevante Informationen: Die Informationen werden in der Regel nach den Elementen des baulich-räumlichen Gefüges strukturiert und somit räumlich kodiert. Beim Umgang mit Gebäuden werden dabei meist bauteil- und raumbezogene Strukturierungen parallel verwendet [Dahlenburg, 1996; Petzold, 2001; von Both u. Kohler, 2004; Thurow, 2004; Schmidt, 1989; Schwarte, 2002]. Dies bedeutet, für einen Entwerfenden spielt das baulich-räumliche Gefüge des zu „beplanenden“ Gebäudes eine zentrale Rolle bei der Orientierung sowohl im Gebäude als auch in zum Gebäude verfügbaren Informationen.

5.5. KONSEQUENZEN FÜR DIE ENTWURFSBEZOGENE INFORMATIONSSUCHE

Zum Abschluss dieses Kapitels kann der Fragekomplex c) zur Informationssuche folgendermaßen beantwortet werden:

Informationssuche stellt einen offenen Prozess dar, der einer Person vor allem dazu dient, einen Zustand der Unwissenheit oder Unsicherheit in einen Zustand des Wissens zu überführen. Suchende interagieren dabei mit dem in einer Informationsumgebung verfügbaren Informationsraum, indem sie – meist in einem iterativen Prozess – Teilmengen von Informationseinheiten bestimmen, diese hinsichtlich ihrer Relevanz für das Informationsproblem untersuchen und ggf. weiter einschränken.

Offener Prozess

Iteratives Bilden und Untersuchen von Teilmengen

Grundsätzlich ist die Suche umso erfolgreicher, je besser mentale Repräsentationen der gesuchten Objekte, des Informationsraums und seiner Schnittstelle mit der tatsächlichen Beschaffenheit, Organisation und Manipulierbarkeit einer Informationsumgebung übereinstimmen.

Mentale

Repräsentationen

Praktisch verwenden Suchende ein breites Spektrum unterschiedlicher Vorgehensweisen, das durch analytische Strategien einerseits und explorative Strategien andererseits begrenzt wird. Analytische Strategien müssen Suchende jeweils bezogen auf eine bestimmte Informationsumgebung erlernen, wobei dieses Wissen nur bedingt auf andere Umgebungen übertragbar ist. Insbesondere im Stadium des Erlernens kanalisieren formale Strategien kognitive Ressourcen stärker als informale Vorgehensweisen. Letztere sind Bestandteil des natürlichen Lernverhaltens und sind damit universell verfügbar. Allerdings sind explorative Strategien nur effizient in einer Umgebung, die über folgende Eigenschaften verfügt:

Suchstrategien

- einen gut strukturierten Informationsraum, nach Möglichkeit mit mehreren Abstraktionsebenen sowie

Explizite Strukturen

- eine Schnittstelle, die die Struktur des Informationsraums für die Suchenden expliziert und gleichzeitig eine direkte Manipulation des Informationsraums erlaubt.

*Präsentation digitaler
Informationsräume*

Zur Interaktion mit einem digital repräsentierten Informationsraum sind Menschen grundsätzlich von der Verfügbarkeit einer angemessenen Nutzerschnittstelle in Form eines oder mehrerer Softwaresysteme angewiesen, die eine für Menschen lesbare und interpretierbare Darstellung der Informationen realisieren.

*Präzisierung der
Zielkriterien*

Zu Beginn dieses Kapitels wurde eine Präzisierung der kategorisch formulierten Anforderungen an entwurfsunterstützende Werkzeuge gefordert. Dies betraf insbesondere den Kontext der entwurfsbezogenen Informationssuche in einer digitalen Bauwerksdokumentation. Anhand der Erkenntnisse dieses Kapitels ist nun eine Präzisierung der zu erfüllenden Zielkriterien wie folgt möglich:

- Verwendung einer für die Entwerfenden vertrauten Strukturierung bzw. Organisation der Bauwerksdokumentation
Diese sollte sich in erster Linie auf das räumliche Gefüge des dokumentierten Bauwerks beziehen.
- Verstärkte Unterstützung explorativer Strategien
Dabei sollte die Schnittstelle die verwendete Strukturierung der Bauwerksdokumentation für Suchende explizit machen und auf Basis dieser Struktur zu interaktiver Exploration einladen.
- Ermöglichen einer kombinierten Anwendung explorativer und analytischer Strategien
Beispielsweise sollten Ergebnisse formaler Anfragen interaktiv erkundet werden können, durch interaktive Vorgehensweisen gebildete Teilmengen sollten durch Spezifikation formaler Suchkriterien weiter eingeschränkt werden können.

6. PROBLEMDEFINITION UND GESAMTKONZEPT

In den vorangegangenen vier Kapiteln wurden die Ausgangssituation sowie die zu berücksichtigenden Randbedingungen für die Strukturierung und die Nutzerschnittstelle einer digitalen Bauwerksdokumentation erläutert. Aufbauend auf diesen Erkenntnissen entwickelt dieses Kapitel eine allgemeine Lösung für die gesuchte Informationsumgebung in ihrer Gesamtheit. Diese besteht aus den drei Ebenen *Daten- bzw. Modellverwaltung*, *Erschließungssystematik* und *Nutzerschnittstelle*. Für die Ebene der Modellverwaltung wird ein variabler Modellverbund nach dem Muster von Peer-DBMS vorausgesetzt. Diese untere Ebene der Informationsumgebung wird um die Erschließungssystematik und die Nutzerschnittstelle ergänzt, die jeweils hinsichtlich ihrer Komponenten und deren Beziehungen beschrieben werden.

*Ausgangspunkt und
Zielsetzung*

Die Entwicklung des Gesamtkonzeptes umfasst dabei folgende Schritte:

Inhalt

- Zusammenfassung der ermittelten Anforderungen
- Gegenüberstellung der Anforderungen und der bestehenden Situation
– Identifikation von Defiziten
- Erläuterung der im Lösungsansatz verwendeten Analogien
- Formulierung eines globalen Lösungsansatzes in Form einer Systemphilosophie.

6.1. BEGRIFFE

Bauwerke bestehen in der Regel aus einzelnen Elementen, die objektiv am realen Bauwerk nachvollziehbar sind. Sie besitzen eine räumliche Dimension und stehen untereinander in Beziehung, so dass räumlich-topologische Strukturen gebildet werden. Gebäude bestehen beispielsweise aus räumlichen Elementen (Raum, Etage, Gebäudeabschnitt etc.) und Bauteilen (Wand, Fenster, Stütze, Balken, Decke etc.), wobei Bauteile Raumgrenzen definieren können, sich innerhalb anderer Bauteile oder in Räumen befinden können und räumliche Elemente oft hierarchische Aggregationsstrukturen bilden. In Bauwerksmodellen werden jeweils aus fachspezifischer Sicht einzelne Eigenschaften dieser Elemente beschrieben und verwaltet. Diese Elemente werden im Folgenden als **Realweltobjekte** bezeichnet, womit auf das reale Vorhandensein eines Objektes innerhalb eines betrachteten Bauwerks verwiesen werden soll, für das jedoch unter Umständen verschiedene rechnerinterne Objekte in verschiedenen fachspezifischen Modellen existieren.

6.2. ANFORDERUNGEN UND ANNAHMEN

Dieser Abschnitt systematisiert zunächst die in den Kapiteln 2 bis 5 erarbeiteten Anforderungen. Parallel werden zusätzliche Annahmen für die zu entwickelnde Lösung getroffen.

Allgemeingültigkeit Der Ansatz eines variablen Modellverbundes als Grundlage der Datenverwaltung und -integration im Bauwesen erhebt einen Anspruch auf Allgemeingültigkeit. Das bedeutet, theoretisch ist ein solcher Modellverbund in allen Phasen der Bauplanung und des Bauwerkslebenszyklus sowie zur Abbildung aller Arten von Bauwerken einsetzbar. Ein Konzept für eine Informationsumgebung, die auf einem solchen Modellverbund aufsetzt, sollte

eine ebensolche Allgemeingültigkeit anstreben. Daher sollte die Systemphilosophie der Informationsumgebung – im Rahmen des betrachteten Kontextes der entwurfsbezogenen Informationssuche – nicht von vornherein auf spezifische bzw. formalisierbare Arten des Informationsbedarfs fokussieren. Informationssuchende sollten zunächst primär bei der Anwendung informaler bzw. explorativer Suchstrategien unterstützt werden, da diese prinzipiell allgemein gültig anwendbar sind. Dabei sollte das Systemkonzept grundsätzlich eine spätere Integration von Werkzeugen vorsehen, die eine Unterstützung analytischer Strategien bieten bzw. in der Lage sind, für formalisierbare Routineaufgaben Informationen ggf. automatisiert bereitzustellen.

Darüber hinaus sind die in den vorangegangenen Kapiteln erarbeiteten Anforderungen wie folgt zu berücksichtigen:

Die Planung im Bestand fordert zunächst allgemein die Verfügbarkeit verlässlicher Informationen zur bestehenden Bausubstanz. Dies bedeutet eine Archivierung *aller* während des Bauplanungsprozesses erstellten Informationen zu einem Bauwerk. Eine Archivierung der zwischen den Planungsbeteiligten ausgetauschten Schnittmenge, beispielsweise auf Basis des IFC-Schemas allein, ist nicht ausreichend.

*Archivierung
planungsrelevanter
Informationen*

Die Auseinandersetzung mit den Fragekomplexen a) bis c) in den Kapiteln 4 und 5 führt zusammenfassend zu folgenden Aussagen:

Beim Entwerfen verwendete Werkzeuge sollten vor allem die Flexibilität des Denkens zulassen bzw. fördern. Ein verstärktes Kanalisieren oder Beanspruchen kognitiver Ressourcen für die Ausführung sekundärer Tätigkeiten ist zu vermeiden. Zu den sekundären Tätigkeiten beim Entwerfen bzw. Entwickeln eines Planungskonzeptes zählt dabei auch das Beschaffen und Interpretieren von Informationen. Daraus resultieren folgende allgemeine Ansprüche an entwurfsunterstützende Werkzeuge:

Entwurfsunterstützung

- Unterstützung informaler, probierender, interaktiver Vorgehensweisen
- Unterstützung von Anschaulichkeit; dies betrifft in erster Linie die im Dialog mit dem Entwerfenden verwendeten Präsentationsformen.

*Entwurfsbezogene
Informationssuche*

Informationsbedarf im Kontext von Entwurfsprozessen entsteht oft spontan und ist nicht vollständig vorhersehbar bzw. formalisierbar. Werkzeuge zur Unterstützung der entwurfsbezogenen Informationssuche müssen es dem Entwerfenden erlauben, die Suche in der bestehenden Bauwerksdokumentation in eigener Verantwortung zu steuern. Für diese Werkzeuge gelten die oben genannten allgemeinen Ansprüche ebenso. Als konkrete Konsequenz ergeben sich folgende Forderungen an die Organisation der Bauwerksdokumentation (BD) und an die Gestaltung der Nutzerschnittstelle (NS):

- (BD) Verwendung einer für die Entwerfenden vertrauten Strukturierung in Orientierung am konkreten Bauwerk, z. B. dem konkreten Raum-Bauteil-Gefüge eines Gebäudes
- (NS) Anschauliche Explikation der Strukturierung bzw. Organisation
- (NS) Kombinierte Unterstützung explorativer und analytischer Suchstrategien
- (NS) Einladen des Nutzers zu interaktiven Vorgehensweisen.

Für die zu realisierende Informationsumgebung werden folgende Annahmen zugrunde gelegt:

*Datenverwaltung in
Form eines flexiblen
Modellverbundes*

Annahme 1. Der verfügbare Informationsraum besteht aus verschiedenen, lose gekoppelten fachspezifischen Modellen nach dem Muster eines Peer-DBMS. Diese werden im Folgenden als *Domänenmodelle* bezeichnet. Die Anzahl sowie die Schemata und Daten der in einem konkreten Fall beteiligten Domänenmodelle ist nicht bekannt und kann über den Verlauf des Bauwerkslebenszyklus variieren. Ein IFC-basiertes Modell kann in einem solchen Verbund einen Peer mit herausgehobener Bedeutung für die Ver-

knüpfung mit anderen Peers bilden. Dies wird jedoch bei der Entwicklung der Lösung nicht vorausgesetzt.

Annahme 2. Die Erstellung und Bearbeitung der Domänenmodelldaten erfolgt durch domänenspezifische Fachapplikationen. Diese stehen nicht notwendigerweise allen Beteiligten zur Verfügung. Dies bedeutet, eine allgemein verfügbare Nutzerschnittstelle zur Suche in der Bauwerksdokumentation kann zur Interpretation und Präsentation der Daten nicht auf Fachapplikationen zurückgreifen.

Präsentation durch Fachapplikationen ist nicht global möglich

Annahme 3. Die einzelnen Peers des Modellverbundes legen jeweils ihre Modellschemata und -daten offen, wobei hinsichtlich der dafür verwendeten Technologie keine Annahmen getroffen werden. Diesbezüglich bietet sich die Verwendung eines Dynamik unterstützenden Modellverwaltungssystems an, da ein solches grundsätzlich den (informierenden) Zugriff auch auf die Modellschemata zulässt. Darüber hinaus wird eine mögliche Bearbeitung der Daten durch Dritte, die während der Suche erfolgen könnte, vernachlässigt. Somit ist eine Überwachung von Änderungen an den Domänenmodellen für die Informationsumgebung zunächst nicht relevant.

Domänenmodelle sind zum Zeitpunkt der Suche unveränderlich

Annahme 4. Während der Planung eines (Neubau-)Projektes wird durch die beteiligten Fachplaner ein Modellverbund mit zunächst unbekanntem Gefüge aufgebaut. Dieser geht nach Abschluss des Projektes in eine digitale Bauwerksdokumentation über. Gegebenenfalls wird der Verbund im Verlauf der Bauwerksnutzung um weitere Domänenmodelle ergänzt, sofern während der Nutzung weitere Daten zum Bauwerk erfasst werden. Für die Planung baulicher Veränderungen steht somit ein Datenbestand in Form „gesammelter“ Domänenmodelle, die ggf. aus unterschiedlichen Lebensphasen des Bauwerkes stammen, zur Verfügung.

Datenbestand in Form „gesammelter“ Domänenmodelle

6.3. DEFIZITE DER BESTEHENDEN SITUATION

Die als gegeben angenommene Situation weist bezüglich der oben genannten Anforderungen mehrere Defizite auf, die bei der Gestaltung der Informationsumgebung zu überwinden sind. Diese betreffen in erster Linie die Organisation des Modellverbundes sowie die Präsentation des gesamten Informationsraums für Informationssuchende. Sie werden im Folgenden jeweils detailliert erläutert.

6.3.1. PROJEKTBEZOGENE ORGANISATION DES MODELLVERBUNDES

*Projektbezogene
Strukturierung* Die Organisation eines variablen Modellverbundes orientiert sich primär an einer projektbezogenen Sichtweise, d. h. die Struktur des Modellverbundes wird in erster Linie durch die fachlich Beteiligten bzw. deren Domänenmodelle bestimmt. Diese Organisationsform ist aufgrund der geforderten Flexibilität und Erweiterbarkeit notwendig, stellt aber einen Nachteil für die Informationssuche in einem so strukturierten Domänenmodellverbund dar.

*Multiple
Repräsentationen* Grundsätzlich beziehen sich alle Domänenmodelle auf denselben Arbeitsgegenstand: ein konkretes Bauwerk. Jede Domäne bildet jedoch nur die für ihre Aufgaben relevanten Merkmale des Arbeitsgegenstandes ab. Unter diesem Blickwinkel sind die Domänenmodellschemata unterschiedliche Modelle eines gemeinsamen übergreifenden Anwendungsbereiches (Bauwesen), die jeweils unterschiedliche fachspezifische Sichten auf diesen Anwendungsbereich beschreiben. Die Daten der verschiedenen Domänenmodelle enthalten somit verschiedene Repräsentationen des gemeinsamen Arbeitsgegenstandes (Bauwerk). Dies bedeutet, dass zu jeweils einem Realweltobjekt (Element des Bauwerks) unterschiedliche Repräsentationen verteilt auf

die verschiedenen Domänenmodelle existieren, die nicht notwendigerweise explizit miteinander verknüpft sind (siehe Abbildung 6.1).

Aufgrund des fehlenden integrierenden Schemas existiert kein zentraler Einstiegspunkt für die Suche in einem derartigen Modellverbund, so dass das Auffinden aller beispielsweise zu einer Wand gehörenden verteilten Repräsentationen schwierig, eventuell sogar unmöglich ist. Abbildung 6.1 verdeutlicht diesen Zusammenhang. Die gestrichelten Linien stellen dabei Verknüpfungen dar, die auf der Ebene des Modellverbundes nicht explizit verfügbar sind. Ein Informationssuchender benötigt diese Verknüpfungen jedoch für die Navigation zu einzelnen domänenspezifischen Repräsentationen eines Realweltobjektes.

*Kein zentraler
Einstiegspunkt*

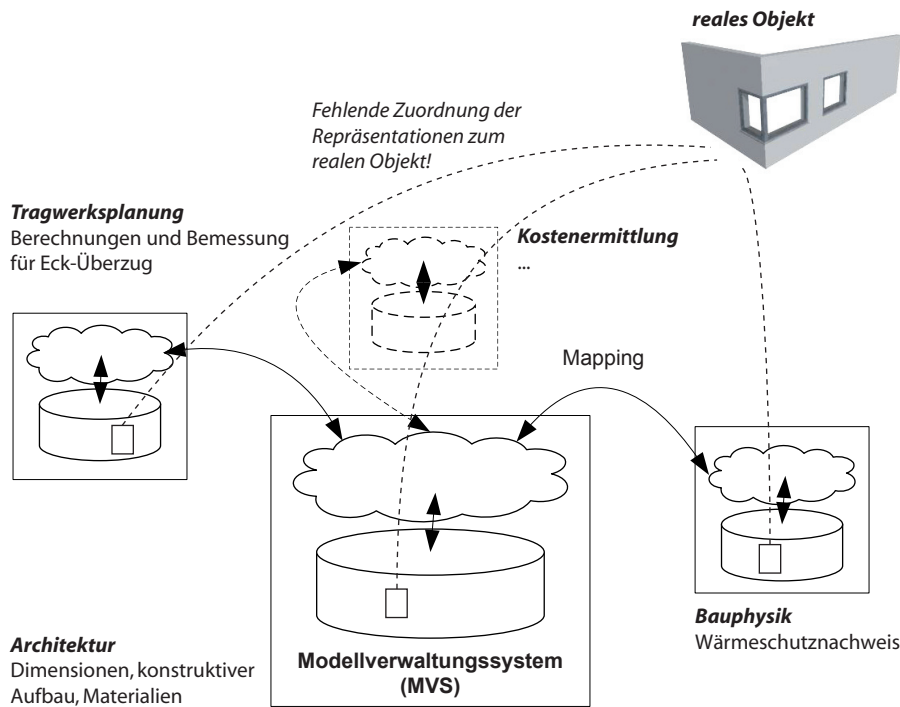


Abbildung 6.1.: Multiple Repräsentationen ohne expliziten Zusammenhang

Dezentrale Struktur Die Modellschemata (lokale Schemata) stehen über Mappings jeweils nur mit einigen Nachbar-Peers in Beziehung. Diese dezentrale Struktur erschwert die Unterstützung analytischer Suche maßgeblich. Prinzipiell können Anfragen nur an einen der Peers gerichtet werden. Falls dieser die Anfrage nicht ausschließlich mit Hilfe seiner Daten beantworten kann, leitet er sie entsprechend bekannter Mappings an Nachbar-Peers weiter. Für Informationssuchende, die mit der Struktur des Modellverbundes nicht vertraut sind, ist es somit schwer, die für ihr Informationsproblem relevanten Peers (Domänenmodelle) zu identifizieren.

Orientierung an der Struktur des konkreten Bauwerks Wie in Kapitel 5 gezeigt wurde, orientieren sich Entwerfende vorrangig an kognitiven Karten bzw. Modellen des konkreten Bauwerks. Dies bedeutet, dass die Organisation des Modellverbundes in Form verteilter fachspezifischer Sichten auf den Arbeitsgegenstand und die mentalen Repräsentationen des Arbeitsgegenstandes, die den Informationssuchenden zur Orientierung dienen, einander nicht entsprechen.

Forderung
Objektbezogene Ordnungsstruktur Daraus ergibt sich die Forderung nach einem Wechsel der primären Ordnungsstruktur innerhalb der Bauwerksdokumentation: Oberhalb des Modellverbundes wird eine Ordnungsstruktur benötigt, die sich am baulich-räumlichen Gefüge des konkreten Bauwerks als Objekt der Planung orientiert. Diese wird im Folgenden als *objektbezogene Ordnungsstruktur* – im Gegensatz zur projektbezogenen Struktur des Modellverbundes – bezeichnet. Diese Struktur sollte die einzelnen Realweltobjekte repräsentieren und damit der Erschließung der in den Domänenmodellen verwalteten Daten für die spätere Suche dienen. Die bestehende Abbildung der Informationen in den Domänenmodellen darf dabei jedoch nicht beeinflusst werden. Dies bedeutet, die Erschließung sollte ohne Modifikationen der Domänenmodellschemata und ohne die Transformation von Domänenmodelldaten auskommen.

6.3.2. GENERISCHER MODELLZUGRIFF

Wie bereits weiter oben erwähnt, kann von der globalen Verfügbarkeit geeigneter Applikationen für die Präsentation aller am Verbund beteiligten Domänenmodelle im Rahmen der Suche nicht ausgegangen werden.

Ein Dynamik unterstützendes Modellverwaltungssystem (MVS) erlaubt zwar den Zugriff auf Modellschemata und -daten mit Hilfe einer generischen Programmierschnittstelle auf Modellierungskonzeptebene (vgl. Abschnitt 3.7). Ein diese Schnittstelle verwendender *Browser* kann jedoch die spezifische Semantik der einzelnen Modelle nicht kennen, da der Modellverbund per Definition variabel hinsichtlich der beteiligten Modelle ist. Daher kann ein solcher *Browser* nur generische alphanumerische Präsentationen der Modellschemata und -daten realisieren. Wie aus Abbildung 6.2 ersichtlich wird, korrespondieren diese Darstellungen nicht mit raumorientierten mentalen Repräsentationen wie kognitiven Karten oder Modellen, die räumliche Situationen abbilden. Die Extraktion von Informationen aus einer solchen Darstellung erfordert vom Entwerfenden einen entsprechend hohen Interpretationsaufwand.¹

Semantik ist nicht transportierbar

Die objektbezogene Ordnungsstruktur soll als Basis für eine Präsentation des baulich-räumlichen Gefüges dienen, die mit den beim Suchenden verfügbaren kognitiven Karten korrespondiert. Dies bedeutet, innerhalb der Ordnungsstruktur müssen die drei in Abschnitt 5.4.2 genannten Eigenschaften der Realweltobjekte beschrieben sein, die für das Einprägen und Wiedererkennen räumlicher Objekte von Bedeutung sind:

Wiedererkennbarkeit von Realweltobjekten

- Identität
- Struktur
- Bedeutung.

¹ Der Leser überlege, wie sich das Ermitteln der *beiden Stützen in der Nord-West-Ecke des Erdgeschosses* aus einer solchen Präsentationsform gestaltet.

6. PROBLEMDEFINITION UND GESAMTKONZEPT

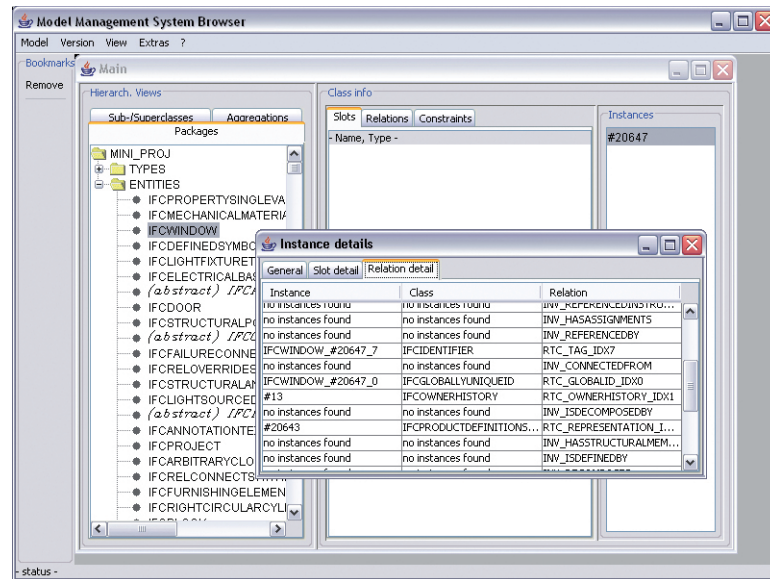


Abbildung 6.2.: Darstellung eines IfcWindow-Exemplars im generischen MVS- Browser

Die Identität eines Realweltobjektes wird maßgeblich durch seine räumliche Dimension und seine Position im Raum bestimmt. Strukturen werden im Wesentlichen durch räumlich-topologische Beziehungen zwischen Realweltobjekten gebildet. Die Bedeutung eines Realweltobjektes kann nur bedingt objektiv erfasst werden. Innerhalb der Ordnungsstruktur sollte sie zumindest durch die Zuordnung eines Elementtyps wie *Raum*, *Etage*, *Wand*, *Stütze* etc. gekennzeichnet werden.

**Forderung explizite
Informationen zur
Visualisierung**

Diese Informationen müssen den in der Ordnungsstruktur enthaltenen „Stellvertretern“ der Realweltobjekte explizit zugeordnet werden. Auf der Basis dieser Informationen sollten verschiedene Zugangsformen zum baulich-räumlichen Gefüge angeboten werden können, wie beispielsweise eine interaktive 3D-Szene, Darstellungen räumlicher Aggregationshierarchien oder Sortierungen anhand von Elementtypen etc.

6.3.3. KOMPLEXITÄT VON BAUWERKSMODELLSCHEMATA

Ein weiteres Problem betrifft nicht die Informationssuche auf der Ebene des Modellverbundes als Ganzes, sondern die Suche innerhalb einzelner Domänenmodelle.

Bauwerksmodelle sind meist so komplex, dass sie für Menschen nicht mehr intuitiv lesbar und erfassbar sind. Katranuschkov u. a. betonen in diesem Zusammenhang die Unterschiede der Denk- und Arbeitsweisen von Menschen (Ingenieuren) und Maschinen (Fachapplikationen) [Katranuschkov u. a., 2003]: Nach Auffassung der genannten Autoren arbeiten Ingenieure mit verschiedenen gedanklichen Abstraktionen eines Sachverhaltes, die jeweils an das bearbeitete Problem angepasst sind. Dieser Auffassung entspricht etwa das, was Dörner als *Komplexionen* und Vester als *Superzeichen* bezeichnen [Dörner, 1976; Vester, 1999]. Parallelen bestehen auch zu den in Abschnitt 5.4.1 beschriebenen mentalen Schemata.

*Verschiedene
gedankliche
Abstraktionen*

Menschen eignen sich durch Erfahrung bestimmte gedankliche Muster an, die es ihnen erlauben, komplexe Sachverhalte nicht als Summe vieler einzelner kleiner Einheiten, sondern als komplexe zusammengehörige Gebilde mit eigener Identität zu erfassen [Dörner, 1976; Vester, 1999].

Gedankliche Muster

Ein Laie mag beispielsweise im Motorraum eines Fahrzeugs nur eine komplexe Anordnung verschiedener Kabel, Schläuche, Flüssigkeitsgefäße und anderer mehr oder weniger kompakter Objekte sehen. Ein KFZ-Mechaniker dagegen wird zwischen Motorblock, Kühlsystem, Bremsanlage, Stromversorgungssystem etc. unterscheiden.

Beispiel

Menschen sind in der Lage, kontextbezogen zwischen verschiedenen Komplexionen ein und desselben Sachverhalts umzuschalten. Das heißt, die Denkmuster sind dynamisch in dem Sinne, dass die für eine Situation nicht benötigten Aspekte ausgeblendet werden können.

Dynamik

Beispiel Zum Starten eines Fahrzeugs genügt die Betrachtung seines Antriebs als eine Einheit. Bei der Fehlersuche an einem nicht startenden Fahrzeug können wahrscheinlich Kühlsystem und Bremsanlage wiederum als Einheiten betrachtet werden. Dagegen wird eine Unterscheidung von Batterie, Zündung, Anlasser usw. hilfreich sein.

Rechnerinterne Repräsentationen Softwareapplikationen dagegen arbeiten auf einem einzigen, immer vollständig und mit allen Details präsenten internen Modell. Dieses muss alle benötigten Informationen in der feinsten geforderten Granularität enthalten [Katranuschkov u. a., 2003]. Daraus resultiert ein wesentliches Dilemma von Bauwerksmodellen: Je größer der Allgemeingültigkeitsanspruch des Modellschemas, d. h. je mehr fachspezifische Aufgaben durch das Schema abgedeckt werden sollen, umso größer werden sein Umfang und seine Komplexität. Große, fachübergreifende Bauwerksmodelle repräsentieren Informationen oft über mehrere Hierarchien verschachtelter Objekte, die für einen Laien nicht mehr intuitiv zu überblicken sind. Abbildung 6.3 zeigt an einem Beispiel aus dem IFC-Schema das Objektgefüge, das die Zuordnung von Materialien zu einem Wandobjekt repräsentiert. Hervorgehoben sind jeweils die Klassen bzw. Objekte, die ein Entwerfender aufgrund gedanklicher Muster intuitiv miteinander assoziieren würde.

Suche in komplexen Schemata Das Auffinden von Informationen innerhalb eines solchen komplexen Modells wird dann zum Problem, wenn keine Softwareapplikationen verfügbar sind, die die Semantik des Schemas interpretieren und so die Informationen entsprechend aufbereiten und präsentieren können. Ohne fundierte Kenntnisse der im Schema definierten Objektstrukturen ist es Informationssuchenden unmöglich, formale Anfragen zielführend zu formulieren. Explorative Vorgehensweisen wären – etwa auf der Basis einer generischen Nutzerschnittstelle wie in Abbildung 6.2 gezeigt – nur in Form eines systematischen Durchsuchens aller Objektbeziehungen über mehrere Hierarchieebenen möglich.

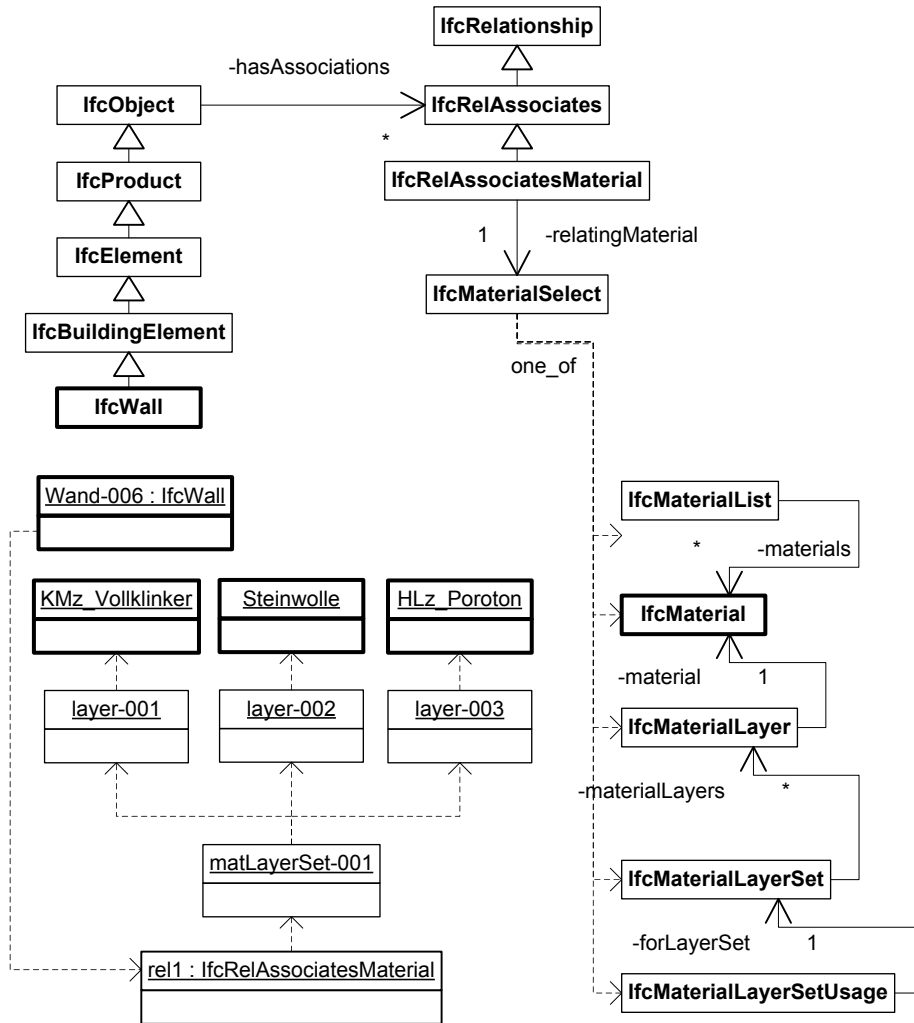


Abbildung 6.3.: Zuordnung von Materialien zu einem Wandobjekt in IFC (UML-Klassen-/Objektdiagramm)

Forderung Zur Unterstützung der Suche innerhalb der potentiell komplexen Strukturen einzelner Domänenmodelle sind daher Mechanismen erforderlich, die von der tatsächlichen Komplexität des Domänenmodellschemas abstrahieren. Die Objektstrukturen des Schemas müssen dabei so aufbereitet werden, dass die wesentlichen Inhalte intuitiv, d.h. auf Basis vorhandener gedanklicher Muster überblickt und erfasst werden können.

Navigationsunterstützung für komplexe Modellschemata

6.3.4. PRÄSENTATIONSFORMEN UND INTERAKTIONSMECHANISMEN

Durch die Nutzerschnittstelle der Bauwerksdokumentation muss die objektbezogene Ordnungsstruktur für die Informationssuchenden explizit präsentiert werden. Zugleich müssen Informationssuchende direkt mit den Darstellungen interagieren können.

Vielfalt und Entwicklung der Techniken Daher stellt sich die Frage, nach welchen Kriterien die Auswahl der durch die Schnittstelle zu realisierenden Präsentations- und Interaktionstechniken zu treffen ist. Grundsätzlich muss davon ausgegangen werden, dass mit der raschen Weiterentwicklung der Hard- und Software auch eine Weiterentwicklung und Verbesserung verfügbarer Präsentations- und Interaktionstechniken erfolgt. Gegenwärtig ist eine Vielzahl verschiedener Techniken bekannt, wobei zunehmend eine Spezialisierung auf verschiedene Kontexte mit erhöhten Anforderungen erfolgt. Exemplarisch seien hier Techniken wie Sprach- und Gestenerkennung, Entwicklungen zum *Wearable Computing* bzw. der gesamte Komplex der *Augmented Reality* genannt. Verschiedene bauwesenspezifische Anwendungen dieser Ansätze sind u. a. in [Petzold u. Bürgy, 2005; Petzold u. a., 2008; Tonn, 2007] beschrieben. Ein Überblick zum derzeitigen Entwicklungsstand von Präsentations- und Interaktionstechniken findet sich beispielsweise in [Ziegler u. Beinhauer, 2007]. Zukünftige Entwicklungen lassen sich nur sehr schwer voraussehen, können jedoch

im Hinblick auf die lange Nutzungsdauer einer Bauwerksdokumentation nicht vernachlässigt werden. Diese Tatsache ist weniger als ein Defizit, sondern vielmehr als eine Unsicherheit bezüglich zukünftiger Veränderungen der gegebenen Situation zu werten.

In diesem Zusammenhang sind die Erkenntnisse des Kapitels 5 von Bedeutung: Die in einer Informationsumgebung verfügbaren Präsentationsformen und Interaktionsmechanismen beeinflussen maßgeblich die Anwendbarkeit und Effizienz einzelner Suchstrategien. Unabhängig von Präsentation und Interaktion muss die Schnittstelle der Informationsumgebung jedoch den Suchenden die Manipulation des Informationsraumes im Zuge der Suche, d. h. das iterative Bilden von Teilmengen erlauben. Daher ist es nicht sinnvoll, die Schnittstelle der Informationsumgebung auf die Realisierung einiger ausgewählter Techniken festzulegen. Stattdessen sollte die Struktur der Schnittstelle eine modulare Erweiterung um die Implementierung zusätzlicher bzw. neuer Präsentations- und Interaktionstechniken bereits im Kern vorsehen. Ebenfalls im Kern der Nutzerschnittstelle anzulegen ist die Funktionalität zur Bildung, Verwaltung und zum Austausch von Teilmengen zwischen einzelnen Präsentations-/Interaktionsmodulen. Auf dieser Basis können später weitere Werkzeuge integriert werden, die ggf. auch ein automatisiertes Bilden von Teilmengen für ausgewählte formalisierbare Problemstellungen oder Routineaufgaben realisieren.

Abhängigkeiten

Forderung *Modulare
Erweiterbarkeit um
neue Präsentations-
und
Interaktionstechniken*

6.4. VERWENDETE ANALOGIEN

In den folgenden Abschnitten werden die Grundprinzipien zweier Ansätze vorgestellt, die in Anteilen als Analogien bei der Entwicklung einer Informationsumgebung für die entwurfsbezogene Informationssuche bzw. einer entsprechenden Systemphilosophie dienen können.

6.4.1. INFORMATIONSUMGEBUNGEN IM DOKUMENTRETRIEVAL

Zur Erschließung unstrukturierter oder für die Suche ungünstig strukturierter Informationsräume bieten explizite, auf einen bestimmten Anwendungsbereich bezogene Erschließungssystematiken, wie sie im Dokumentretrieval verwendet werden, einen geeigneten Ansatz. Elektronische Retrievalsysteme ermöglichen auf Basis dieser Erschließungssystematiken den Zugriff auf den Informationsraum.

AUFBAUPRINZIP EINER RETRIEVALUMGEBUNG

Zugriff auf Dokumente Im Informations- und Dokumentationswesen dienen elektronische Retrievalsysteme in Kombination mit so genannten *Dokumentationssprachen* der Erschließung und dem gezielten Zugriff auf eine Menge von Dokumenten, beispielsweise wissenschaftlichen Texten [Burkart, 1997; Laux, 1997; Manecke, 1997]. Dabei wird durch die Verknüpfung der Dokumente mit einzelnen Begriffen der Dokumentationssprache ein für die Art und den Zweck des Zugriffs strukturierter Informationsraum geschaffen. Dieser Arbeitsschritt wird als Klassieren bzw. Indexieren bezeichnet [Knorz, 1997]. Das Retrievalsystem verwendet wiederum die Dokumentationssprache, um die Dokumente zu ermitteln, die mit einem oder mehreren Begriffen verknüpft wurden.

Ebenenstruktur Wie in Abbildung 6.4 dargestellt, können somit drei Ebenen innerhalb einer Informationsumgebung unterschieden werden:

- die Menge verfügbarer Dokumente
als unstrukturierter oder für die Suche ungünstig strukturierter Informationsraum,

- die Dokumentations-sprache
als Struktur zur Erschließung des Informationsraums und als vermittelnde Schicht zum Retrievalsystem sowie
- das Retrievalsystem
als Nutzerschnittstelle zum gezielten Zugriff auf einzelne Dokumente des Informationsraums unter Verwendung der Dokumentations-sprache.

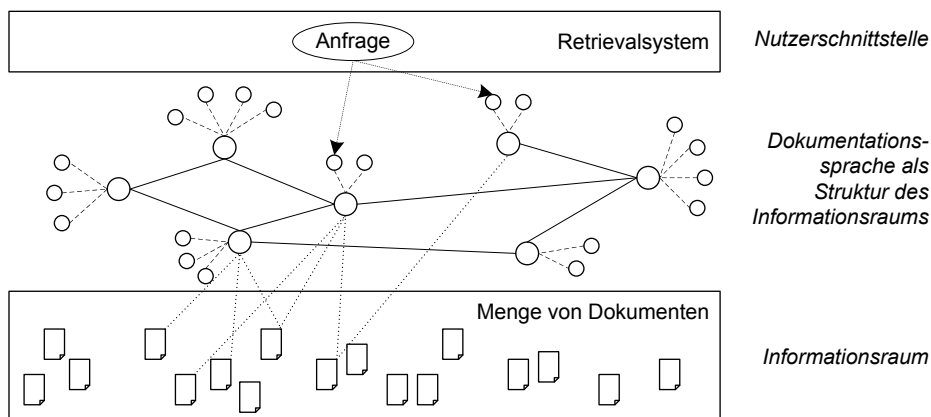


Abbildung 6.4.: Ebenen einer Informationsumgebung zum Dokumentretrieval

DOKUMENTATIONSSPRACHEN

Unter einer *Dokumentationssprache*² wird im Informations- und Dokumentationswesen eine explizite Beschreibung der für einen Anwendungsbereich

Inhalt und Zweck

² Die Bezeichnung *Dokumentationsstruktur* wäre treffender, da lediglich eine Menge gültiger Symbole (Begriffe) und struktureller Beziehungen definiert wird, jedoch keine syntaktischen Regeln, wie dies bei natürlichen oder formalen Sprachen der Fall ist. Die Verwendung der Bezeichnung erfolgt hier dennoch in der durch das Informations- und Dokumentationswesen zugewiesenen Bedeutung.

gültigen Begriffe und deren Beziehungen verstanden, die letztlich der Systematisierung einer umfangreichen Menge von Dokumenten zu diesem Anwendungsbereich dient. Im Wesentlichen zählen dazu sowohl *Klassifikationssysteme* als auch *Thesauri* [Burkart, 1997; Knorz, 1997; Manecke, 1997; Taube, 1998].

Kontextbezug Dokumentations Sprachen werden in der Regel in einem manuellen Prozess für ein konkretes Anwendungsgebiet und einen bestimmten Verwendungszweck³ entwickelt [Knorz, 1997]. Sie repräsentieren somit einen Teil des für ihren Anwendungsbereich gültigen Wissens in Form explizit definierter Begriffe und der Darstellung begrifflicher Zusammenhänge.

Begriffe und Bezeichnungen Bei der Entwicklung einer Dokumentations Sprache kommen zwei wesentliche Prinzipien zur Anwendung:

- die *Begriffsbildung* und
- die *Externalisierung* von Begriffen durch *Bezeichnungen*.

Begriffsbildung bezeichnet den kognitiven Prozess der Zusammenfassung ähnlicher Inhalte zu einer Kategorie (siehe S. 116). Begriffe müssen durch die Zuordnung äquivalenter sprachlicher Bezeichnungen externalisiert werden, um für die Dokumentations Sprache explizit nutzbar zu sein. Abbildung 6.5 stellt diesen Zusammenhang dar.

Identifikatoren Aufgrund der Ambiguität⁴ der natürlichen Sprache erhalten die einzelnen Begriffe meist einen Identifikator, der sie innerhalb des Gültigkeitsbereiches der Dokumentations Sprache eindeutig kennzeichnet. Bei Klassifikationssystemen spricht man in diesem Zusammenhang von *Notationen* [Manecke, 1997]. Dies sind nach spezifischen Regeln zusammengesetzte künstliche Zeichenfolgen, die zugleich die Position der Klasse (des Begriffs) im hierarchischen Gesamtsystem kennzeichnen. In Thesauri werden die Identifikatoren

³ d. h. die Art und Weise der späteren Suche

⁴ Mehrdeutigkeit

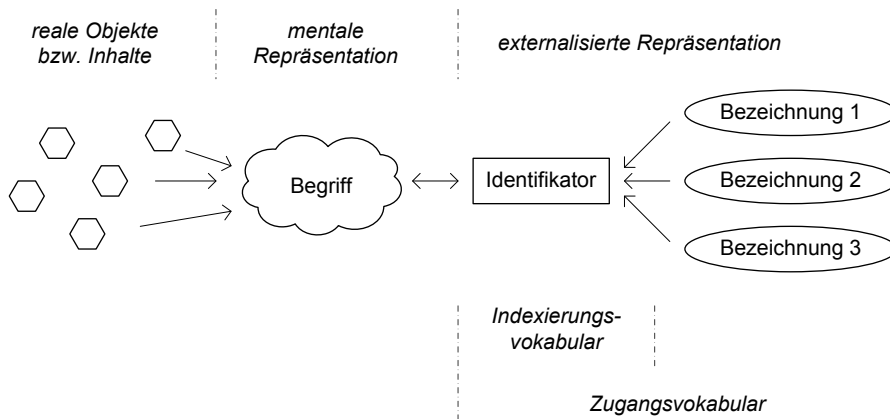


Abbildung 6.5.: Begriffsbildung und Zuordnung von Bezeichnungen in einer Dokumentationsprache

als *Deskriptoren* bezeichnet [Burkart, 1997]. Sie sind in diesem Fall natürlichsprachliche Bezeichnungen, die aus der Menge der für einen Begriff erfassten Synonyme gewählt werden. Das Indexieren erfolgt ausschließlich unter Verwendung der Identifikatoren. Die erfassten synonymen Bezeichnungen stehen jedoch ebenfalls als Zugangsvokabular zur Verfügung. Sie verweisen jeweils auf den Identifikator des zugehörigen Begriffs, anhand dessen zugeordnete Dokumente gefunden werden können.

Die Begriffe einer Dokumentationsprache werden durch explizit in der Sprache beschriebene Relationen zueinander in Beziehung gesetzt. Dieses Prinzip entspricht dem Modell der semantischen Netzwerke (vgl. S. 116). Wesentliche Arten interbegrifflicher Beziehungen sind dabei Teil-Ganzes-Beziehungen, Unterbegriffs-Overbegriffs-Beziehungen, Äquivalenzbeziehungen sowie allgemeine, nicht näher spezifizierte Assoziationen [Burkart, 1997].

Relationen zwischen Begriffen

6.4.2. OFFENE HYPERMEDIA-SYSTEME

Offene Hypermedia-Systeme bieten einen geeigneten technischen Ansatz, um bestehende Inhalte im Nachhinein durch Hypermedia-Strukturen zu verknüpfen und diese Strukturen Nutzern zur Navigation zwischen den Inhalten bereitzustellen.

BEGRIFFE

Nicht-lineare Organisation **Hypertext** bzw. **Hypermedia** stellt einen durch elektronische Systeme möglich gewordenen Ansatz zur Repräsentation von Informationen dar, der über Text als kohärente lineare Einheit hinausgeht – daher die Bezeichnung *Hypertext* [Kuhlen, 1997]. Einzelne in sich geschlossene Informationseinheiten werden durch so genannte *Hyperlinks* untereinander verknüpft und so in Form eines nicht-linearen Netzwerks organisiert. Dieses Konzept der Vernetzung von Informationseinheiten wurde bereits in den 40er Jahren formuliert [Bush, 1945].

Verwendung der Begriffe Die meisten Autoren unterscheiden nicht explizit zwischen Hypertext und Hypermedia, da Hypertext-Dokumente meist außer in Textform repräsentierten Informationen andere Formate wie Bild, Ton oder Animationen einbinden [Kuhlen, 1997; Taube, 1998; Marchionini, 1997]. In diesem Sinne werden im Folgenden die Bezeichnungen Hypertext und Hypermedia synonym verwendet.

Verknüpfungen sind für System und Suchenden explizit Das für Hypermedia Spezifische ist jedoch nicht ausschließlich die Verwendung vernetzter Strukturen, sondern darüber hinaus die Art und Weise ihrer Repräsentation: Verknüpfungen zwischen einzelnen Informationseinheiten werden nicht nur systemintern repräsentiert, wie dies im Prinzip auch beim Indexieren mit Hilfe von Dokumentationssprachen der Fall ist. Sie werden gleichzeitig für den Benutzer explizit gemacht, indem sie als

Befehlsschaltflächen in die am Bildschirm dargestellten Inhalte integriert werden [Taube, 1998]. Erst durch diese Tatsache ist Navigationsverhalten beim Lesen von Hypertext-Dokumenten überhaupt möglich.

Der Begriff **offenes Hypermedia-System**⁵ (OHS) bezeichnet im Allgemeinen Komponenten einer Zwischenschicht, die Applikationen in einer Systemumgebung Funktionalität zum Erzeugen, Modifizieren und Löschen von Hyperlinks sowie zum Traversieren dieser Links bereit stellen. Derartige Funktionalitäten werden auch als *Hypermedia-Dienste*⁶ bezeichnet [Wiil, 1997].

*Offenes
Hypermedia-System*

Die folgenden Erläuterungen zu den Grundlagen von OHS beziehen sich auf [Wiil, 1997; Nürnberg u. Leggett, 1997; Anderson u. a., 1997, 1999].

Ein elementares Grundprinzip von OHS ist die getrennte Verwaltung von Inhalten und Strukturen. Unter **Inhalten** werden allgemein sämtliche von unterschiedlichen Applikationen der Systemumgebung erzeugten und verwalteten Informationseinheiten verstanden. Informationseinheiten können dabei in Form von Dateien verwaltete Dokumente, Teile dieser Dokumente oder auch Datensätze in Datenbanken sein. **Hypermedia-Strukturen** werden durch explizit zwischen diesen Informationseinheiten definierte Hyperlinks gebildet und separat von den Inhalten durch ein OHS verwaltet und bereitgestellt.

*Trennung von
Inhalten und
Strukturen*

Im Gegensatz zu geschlossenen oder monolithischen Hypermedia-Systemen legen OHS lediglich das Datenformat zur Verwaltung der durch Links gebildeten Strukturen fest. Parallel definieren sie Schnittstellen, über die beliebige Applikationen die Dienste zur Erstellung, Manipulation und Verwendung der Links aufrufen können. Hinsichtlich dieser Client-Applikationen sowie der von ihnen bearbeiteten Datenformate trifft ein OHS keinerlei Annahmen. Applikationen können somit Inhalte in unterschiedlichen Formaten

*Offen vers.
monolithisch*

⁵ im Englischen *open hypermedia system*

⁶ im Englischen *hypermedia services*

*Überlagerung von
Inhalten und
Strukturen*

außerhalb des OHS erzeugen und verwalten, was die Integration bestehender Inhalte und Applikationen in die Hypermedia-Umgebung ermöglicht. Monolithische Systeme dagegen legen die Datenformate sowohl zur Verwaltung von Strukturen als auch zur Verwaltung von Inhalten fest und erlauben so nur einer begrenzten Menge von Applikationen die Partizipation in der Hypermedia-Umgebung. Inhalte und Strukturen werden erst zum Zeitpunkt der Anzeige innerhalb von Applikationen überlagert, d. h. während des Ladens einer Informationseinheit ermitteln Applikationen jeweils vom OHS die für diese Informationseinheit definierten Anker und Links und heben diese innerhalb der Darstellung hervor. Um die dafür benötigten Dienste des OHS verwenden zu können, müssen die Applikationen entsprechend angepasst werden.

REFERENZARCHITEKTUR FÜR OHS

Schichten

In [Grønbæk u. Wiil, 1997; Anderson u. a., 1999] wird eine Referenzarchitektur für OHS beschrieben. Sie besteht aus drei Schichten, die jeweils eine oder mehrere Komponenten enthalten können, wie in Abbildung 6.6 gezeigt.

Funktionalität

Die Infrastrukturschicht (*hypermedia database layer*) übernimmt die persistente Speicherung von Inhalten und Strukturen, wird von den Autoren jedoch nicht näher spezifiziert. Die darüber liegende mittlere Schicht (*open hypermedia framework layer*) implementiert Hypermedia-Dienste und stellt diese den Applikationen der obersten Schicht (*content handler layer*) zur Verfügung. Definierte Schnittstellen zwischen den Schichten regeln die Kommunikation zwischen den Komponenten unterschiedlicher Schichten. Auf diese Weise kann jede der Schichten um neue Komponenten erweitert werden, sofern diese die gegebenen Schnittstellen implementieren.

Die Schnittstelle zwischen der oberen Applikationsschicht und den Hypermedia-Diensten (*Content Handler Interface CHI*) wurde in Form eines

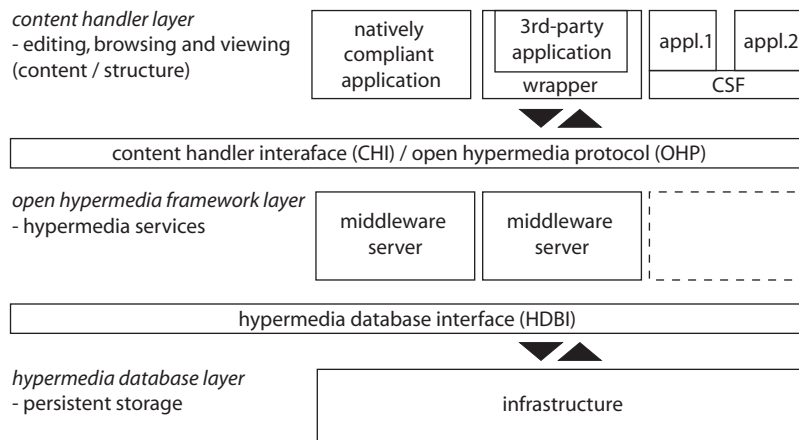


Abbildung 6.6.: Referenzarchitektur für OHS nach [Anderson u. a., 1999]

Protokolls (*Open Hypermedia Protocol OHP*) spezifiziert. Es definiert jeweils eine Menge gültiger Nachrichten für

- Aufrufe vom OHS an eine Client-Applikation,
- Rückmeldungen der Client-Applikation,
- Aufrufe einer Client-Applikation an das OHS und
- Rückmeldungen des OHS.

Vgl. hierzu [Anderson u. a., 1997, 1999].

EIN DATENMODELL FÜR OFFENE HYPERMEDIA-STRUKTUREN

Unter dem Namen *Fundamental Open Hypertext Model (FOHM)* wird in [Millard u. a., 2000; Weal u. a., 2001] ein Datenmodell für die Verwaltung von Hypermediastrukturen beschrieben. Abbildung 6.7 zeigt die elementaren Konzepte dieses Modells anhand eines UML-Klassendiagramms.

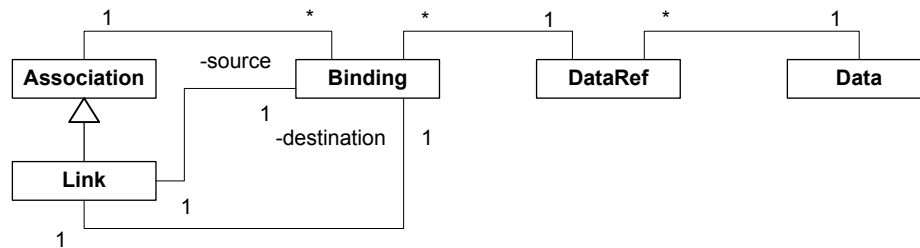


Abbildung 6.7.: UML-Klassendiagramm der FOHM-Konzepte

Assoziationen Exemplare von **Association** repräsentieren in FOHM alle Arten von Strukturen, die Beziehungen zwischen Informationseinheiten beschreiben. Dies können Hyperlinks sein, die jeweils zwei Informationseinheiten verknüpfen, aber ebenso Verknüpfungen mehrerer Informationseinheiten zu Listen oder Sets.

Daten/Inhalte **Data** steht als Platzhalter für jede Art von Informationseinheit. Da die Verwaltung der Informationseinheiten bzw. Inhalte nicht Gegenstand von OHS ist, erfolgt keine weitergehende Spezifikation.

Referenzen **Association**-Exemplare referenzieren Informationseinheiten nicht direkt, sondern über Exemplare von **DataRef**. **DataRef**-Objekte definieren dabei Zeiger auf Informationseinheiten in ihrer Gesamtheit oder auf ausgezeichnete Teile innerhalb von Informationseinheiten.

Bindungen Exemplare von **Binding** verknüpfen jeweils **DataRef**-Objekte und Assoziationen. Jeder Assoziationstyp kann individuell Anzahl und Bedeutung der jeweils für diesen Typ erforderlichen Bindungen definieren. So enthält beispielsweise jedes **Link**-Objekt als gerichtete Verknüpfung zwischen zwei Informationseinheiten genau eine Bindung für den Startpunkt und genau eine Bindung für den Endpunkt der Verknüpfung.

6.5. DAS VIRTUELLE BAUWERK ALS INFORMATIONSUMGEBUNG

Für die angestrebte Lösung wird davon ausgegangen, dass das Finden des Planungskonzeptes Aufgabe des Entwerfenden bleibt. Die Informationsumgebung soll dabei jedoch insofern unterstützen, dass sie den Akteuren einen flexiblen, anschaulichen Zugang zu den zur bestehenden Bausubstanz verfügbaren Informationen bietet. Dafür muss sowohl die Erschließung des Modellverbundes als auch die Präsentation des Informationsraums für die Entwerfenden realisiert werden.

*Ausschließlich
Bereitstellung von
Informationen*

In Analogie zur Ebenenstruktur einer Retrievalumgebung wird dafür eine aus drei Schichten bestehende Systemstruktur vorgeschlagen, wie sie in Abbildung 6.8 schematisch dargestellt ist. Parallel zeigt Abbildung 6.9 die im Folgenden verwendeten Bezeichnungen für die einzelnen Schichten und Komponenten des Systemkonzeptes.

Systemkonzept

Das Äquivalent zur Dokumentationssprache bildet die *Erschließungsstruktur* als eine Menge unabhängig vom Modellverbund repräsentierter Strukturen, die den Zugang zu den Daten des Modellverbundes in Korrespondenz mit kognitiven Karten und gedanklichen Mustern ermöglichen. In Analogie zum Ansatz von OHS werden diese Strukturen separat von den Inhalten des Modellverbundes definiert und verwaltet. Die Erschließungsstruktur umfasst dabei zwei Anteile: die objektbezogene Ordnungsstruktur und Eigenschaftslinks.

Erschließungsstruktur

Die *objektbezogene Ordnungsstruktur* repräsentiert jeweils das spezifische baulich-räumliche Gefüge des Bauwerks. Dies erfolgt durch Anlegen von *Stellvertretern* als eindeutige Abbildungen der navigationsrelevanten Realweltobjekte. Bei Gebäuden sind dies in der Regel einzelne räumliche Einheiten (Räume, Raumgruppen, Geschosse etc.) und die raumbildenden Bau-

*Objektbezogene
Ordnungsstruktur*

6. PROBLEMDEFINITION UND GESAMTKONZEPT

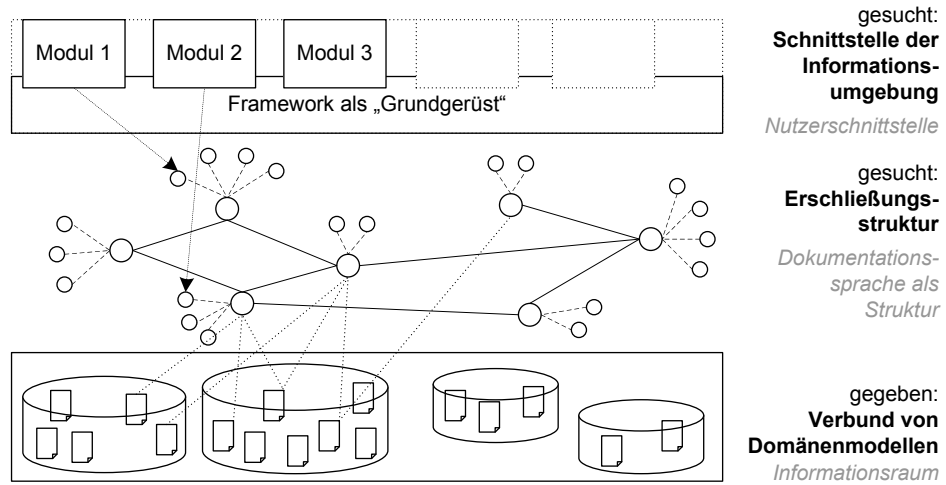


Abbildung 6.8.: Vorgeschlagene Systemstruktur in Analogie zum Aufbau einer Retrievalumgebung

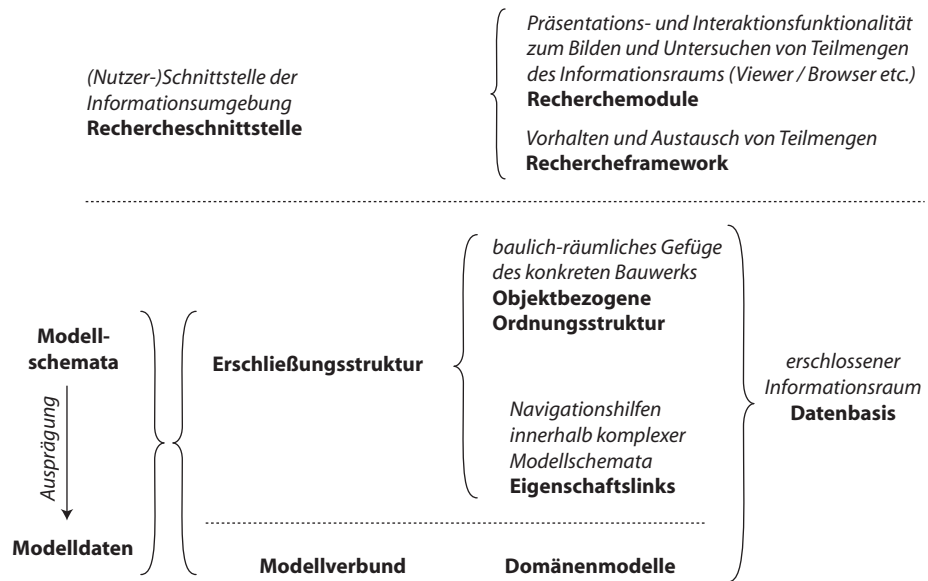


Abbildung 6.9.: Bezeichnungen für die Komponenten der Informationsumgebung (fett) und deren Aufgaben (kursiv)

teile. Den Stellvertretern werden jeweils beschreibende Informationen zur Identität und Bedeutung des Realweltobjektes sowie strukturbeschreibende Beziehungen zu anderen Realweltobjekten (deren Stellvertretern) zugeordnet. Auf diese Weise wird in der digitalen Repräsentation des Bauwerks eine Entsprechung zu den kognitiven Karten, die Entwerfende zur Orientierung benutzen, geschaffen. Die im Modellverbund verwalteten Informationen werden mit den Stellvertretern der Realweltobjekte in Beziehung gesetzt, wodurch multiple Repräsentationen eines Objektes über einen zentralen Einstiegspunkt zugänglich werden (siehe Abbildung 6.10, linke Seite).

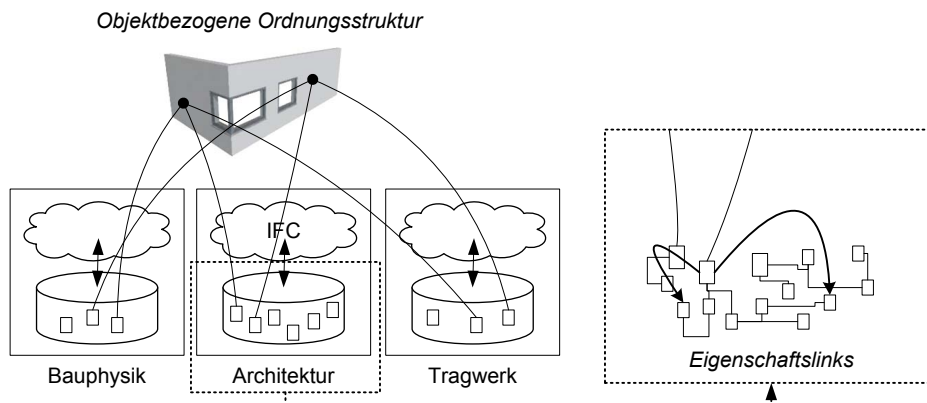


Abbildung 6.10.: Objektbezogene Ordnungsstruktur und Eigenschaftslinks

Den zweiten Teil der Erschließungsstruktur bilden Mechanismen zur Unterstützung der Navigation innerhalb einzelner komplexer Domänenmodelle, d. h. zur Unterstützung beim Traversieren komplexer Objektstrukturen. Formale Anfragen können als Beschreibung von Regeln zum Traversieren und zur Auswertung komplexer Objektstrukturen dienen. Werden entsprechend vordefinierte Anfragen durch in der Erschließungsstruktur definierte Verknüpfungen mit einzelnen Konzepten eines Domänenmodellschemas in Beziehung gesetzt, können auf Basis dieser Verknüpfungen Informationen während der Präsentation durch die Nutzerschnittstelle aufbereitet bzw. zu-

Eigenschaftslinks

sammengefasst werden (siehe Abbildung 6.10, rechte Seite). Diese Art von Verknüpfungen wird im Folgenden als *Eigenschaftslink* bezeichnet.

Erschlossener Informationsraum als Basis der Bauwerksdokumentation Die Erschließungsstruktur wird ebenso wie die Domänenmodelle mit Hilfe des objektorientierten Paradigmas modelliert. Dies bedeutet, es ist ein entsprechendes konzeptuelles Schema zu entwickeln, dessen Ausprägungen jeweils die Erschließungsstruktur für ein konkretes Bauwerk/eine Bauwerksdokumentation bilden. Durch explizite Zuordnungen zwischen den Daten des Modellverbundes und der Erschließungsstruktur entsteht ein für die entwurfsbezogene Informationssuche erschlossener Informationsraum als Datenbasis der Informationsumgebung.

Rechercheschnittstelle Als Äquivalent zum Retrievalsystem benötigt die Informationsumgebung eine Nutzerschnittstelle, die dem Informationssuchenden die Interaktion mit dem digital repräsentierten Informationsraum ermöglicht. Für diese Schnittstelle wird im Folgenden die Bezeichnung *Rechercheschnittstelle* verwendet.

Erweiterbarkeit Eine der wichtigsten Anforderungen an die Rechercheschnittstelle ist die Erweiterbarkeit um zusätzliche Interaktions- und Präsentationstechniken in Form weiterer Module. Dabei muss sichergestellt werden, dass die neu hinzugefügten Module mit den bereits vorhandenen kommunizieren und interagieren können. An diesem Punkt stellt sich ein entscheidendes Problem: Die Rechercheschnittstelle muss entwickelt werden, ohne dass zum Zeitpunkt der Entwicklung alle später im konkreten Anwendungsfall beteiligten Komponenten (Module) bekannt sind. Dies ist nur mit Hilfe der drei folgenden Prinzipien sinnvoll lösbar:

Modularer Aufbau

- Zerlegung der Schnittstelle in einzelne Module
- Kapselung der variablen Prinzipien (Darstellung und Benutzerinteraktion)

- Bereitstellung der gemeinsam genutzten Funktionalität in Form eines Skeletts der Anwendung, das Schnittstellen für später einzufügende Module definiert.

Vor diesem Hintergrund steht die Zerlegung der Recherveschnittstelle in zwei Ebenen:

- das Rechercheframework und
- eine variable Menge verschiedener Arten von Recherchemodulen.

Das *Rechercheframework* bildet die erste Ebene der Schnittstelle. Es realisiert *Basisfunktionalität* zur Manipulation des Informationsraums und zur Regelung der Kommunikation zwischen einzelnen Recherchemodulen. Dazu zählen vor allem die temporäre Verwaltung der vom Informationssuchenden gebildeten Teilmengen und der Austausch dieser Teilmengen zwischen den Modulen. Diese Basisfunktionalität wird für alle später in Form von Plug-Ins hinzugefügten Recherchemodule bereitgestellt.

Rechercheframework

Basisfunktionalität

Benutzer interagieren nicht direkt mit dem Framework, sondern mit einzelnen *Recherchemodulen*. Diese Module implementieren jeweils verschiedene Interaktions- und Präsentationstechniken. Bei der Umsetzung der einzelnen Module kann jeweils entschieden werden, ob eher explorative oder verstärkt analytische Strategien (z. B. für einzelne Routineaufgaben) unterstützt werden sollen. Die Module sind austauschbar und können beliebig in das vom Framework definierte Grundgerüst eingefügt werden. Dafür müssen sie jedoch die vom Framework definierten Schnittstellen implementieren, um mit diesem und somit auch mit anderen Modulen kommunizieren zu können.

Recherchemodule

Auf diese Weise sind die Informationssuchenden frei in der Wahl der im konkreten Fall benutzten Recherchemodule. In Abhängigkeit von der Art und Anzahl der verfügbaren Recherchemodule ist so eine sehr flexible Informationssuche möglich. Darüber hinaus können jederzeit weitere

Flexible Nutzbarkeit

Recherchemodule und damit Implementierungen neuer Präsentations- und Interaktionstechniken ergänzt werden.

6.6. ABGRENZUNG DES BETRACHTETEN SACHVERHALTES

Bei der weiteren Konkretisierung des als grundlegende Systemphilosophie formulierten Lösungsansatzes werden einige Aspekte des Problemkontextes zunächst ausgeklammert. Dazu zählen unter anderem:

- die Bearbeitung der im Modellverbund verwalteten Daten,
- Prozesse zur Generierung von Erschließungsstruktur-Daten,
- verschiedene rechtliche Fragestellungen sowie
- die Entwicklung und Implementierung von Interaktions- und Präsentationstechniken.

*Fokus auf explorative
Suche*

Darüber hinaus konzentriert sich die Konzeption und Beschreibung der einzelnen Systemkomponenten für die Erschließungsstruktur und das Framework zunächst vorrangig auf Mechanismen zur Unterstützung explorativer Suchstrategien. Eine Umsetzung des Matching-Paradigmas innerhalb der Recherneschnittstelle einschließlich der Entwicklung entsprechender Algorithmen soll in den weiteren Betrachtungen nicht im Mittelpunkt stehen. Dies steht zwar formal im Gegensatz zur Forderung nach einer integrierten Unterstützung analytischer und explorativer Strategien. Dennoch soll der Unterstützung explorativer Strategien bei der folgenden Spezifikation der zentralen Komponenten der Vorrang gegeben werden, da diese eine spielerische Herangehensweise an die Konzeptentwicklung bzw. das Entwerfen im Kontext bestehender Bausubstanz stärker fördern als analytische Strategien.

BEARBEITUNG DER VERWALTETEN DATEN

Zunächst soll entsprechend der *Annahme 3* (siehe S. 125) davon ausgegangen werden, dass während der Informationssuche keine Bearbeitung der im Modellverbund enthaltenen Daten erfolgt. Zugriffe während der Informationssuche erfolgen ausschließlich lesend. Die Entwerfenden arbeiten nicht mit dem virtuellen Bauwerk im Sinne einer Experimentierumgebung. Darüber hinaus wird auch eine Bearbeitung der Modelldaten durch andere Fachplaner während einer Rechercesitzung vernachlässigt. Entsprechende Mechanismen zur Unterstützung synchroner/asynchroner Kooperation verschiedener Beteiligten, wie beispielsweise Überwachung und Notifikation von Änderungen, Versionierung etc. sind nicht Gegenstand der Arbeit.

GENERIERUNG DER ERSCHLIESSUNGSSTRUKTUR-DATEN

Die Arbeit konzentriert sich im Folgenden auf die Entwicklung eines konzeptuellen objektorientierten Schemas zur Repräsentation der Erschließungsstruktur sowie zur Repräsentation von Zuordnungen zwischen Elementen der Erschließungsstruktur und Domänenmodelldaten. Für die Entwicklung des Rechercheframeworks wird anschließend die Verfügbarkeit entsprechender Daten als Ausprägungen des Schemas angenommen. Möglichkeiten einer automatisierten oder halbautomatischen Überführung der benötigten Informationen aus bereits verfügbaren Domänenmodelldaten sollen hier nicht untersucht werden. Ferner wird auch der Prozess der Definition/Erstellung von Zuordnungen zwischen Erschließungsstruktur und Domänenmodelldaten nicht weiter betrachtet. Diese Aspekte werden als eine zweite Stufe der Gesamtlösung angesehen und sind daher weiterführenden Arbeiten vorbehalten.

RECHTLICHE FRAGESTELLUNGEN

Zweifellos berührt ein Ansatz, der eine vollständige Archivierung aller Planungsinformationen und damit deren Bereitstellung für einen potentiellen Kreis von Nutzern, der nicht mehr ausschließlich die Planungsbeteiligten umfasst, auch die Frage von Urheberrechten. Diese soll jedoch hier vernachlässigt werden. Ebenso wird von einer Untersuchung zu Fragen der Zuordnung, Verwaltung und Prüfung von Zugriffsrechten an einzelne Nutzer oder Nutzergruppen abgesehen.

INTERAKTIONS- UND PRÄSENTATIONSTECHNIKEN

Eine begründete Auswahl, Weiterentwicklung und Implementierung von Interaktions- und Präsentationstechniken soll nicht Gegenstand der Arbeit sein. Gerade aufgrund des zu berücksichtigenden Entwicklungsfortschritts ist die modulare Erweiterbarkeit diesbezüglich elementarer Bestandteil der Systemphilosophie. Die Arbeit konzentriert sich stattdessen auf die Frage der Organisation und Koordination einer beliebigen Menge einzelner Recherchemodule sowie die Kommunikation und Interaktion zwischen diesen Modulen. Denn nur auf dieser Basis ist eine flexible Erweiterbarkeit überhaupt möglich.

Abbildung 6.11 zeigt das in diesem Kapitel als Systemphilosophie formulierte Gesamtkonzept der Informationsumgebung. Die beiden Arbeitsschwerpunkte der folgenden Kapitel sind dabei hervorgehoben: Auf der einen Seite steht die Entwicklung eines konzeptuellen objektorientierten Schemas zur Abbildung konkreter Erschließungsstrukturen in Kapitel 7. Auf der anderen Seite wird in Kapitel 8 das Rechercheframework einschließlich der notwendigen Schnittstellen zur Integration von Recherchemodulen als ein objektorientiertes System spezifiziert.

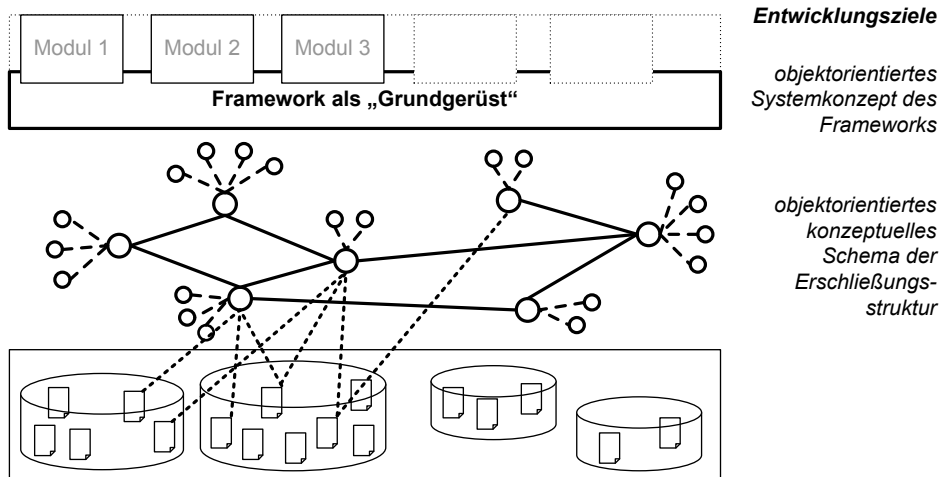


Abbildung 6.11.: Abgrenzung der Arbeitsschwerpunkte

6. PROBLEMDEFINITION UND GESAMTKONZEPT

7. ERSCHLIESSUNG DES INFORMATIONSRRAUMS

Gegenstand dieses Kapitels ist die Entwicklung eines konzeptuellen Schemas zur Verwaltung der Erschließungsstruktur (ES) einer digitalen Bauwerksdokumentation. Dieses wird im Folgenden als Erschließungsstruktur-Schema (**ES-Schema**) bezeichnet. Die Erschließungsstruktur für eine konkrete Bauwerksdokumentation stellt eine Ausprägung dieses Schemas dar. Für diese wird die Bezeichnung Erschließungsstruktur-Daten (**ES-Daten**) verwendet. *Zielsetzung*

Die vorgeschlagene Lösung stützt sich wesentlich auf den in offenen Hypermedia-Systemen (vgl. Abschnitt 6.4.2) verfolgten Ansatz der getrennten Verwaltung von Inhalten und Linkstrukturen, die Nutzern bei der Navigation zwischen Inhalten dienen. Mit diesem Prinzip bleibt der Modellverbund unabhängig von der Erschließungsstruktur. *Ansatz*

Entsprechend der im vorangegangenen Kapitel getroffenen Abgrenzung steht hier zunächst ausschließlich die Entwicklung und Beschreibung der benötigten Datenstrukturen im Mittelpunkt. Prozesse zur Erstellung und Bearbeitung der Daten sowie entsprechende Verantwortlichkeiten werden dabei nicht betrachtet. Diese Aspekte sowie Ansätze zur Entwicklung entsprechender Lösungen werden jedoch zum Abschluss des Kapitels bzw. in Kapitel 9 diskutiert. *Abgrenzung*

- Inhalt* Im Einzelnen umfasst dieses Kapitel folgende Schwerpunkte:
- Präzisierung der Lösungsansätze für die objektbezogene Ordnungsstruktur sowie die Eigenschaftslinks
 - Entwicklung eines konzeptuellen, objektorientierten Schemas (ES-Schema) zur Verwaltung der Erschließungsstruktur
 - Vorschlag zur formalen, rechnerinternen Repräsentation der Erschließungsstruktur (ES-Schema und -Daten).

7.1. INFORMATIONSCONTAINER UND EIGENSCHAFTSLINKS

Dieser Abschnitt erläutert die vorgeschlagenen Lösungsansätze zum Aufbau der objektbezogenen Ordnungsstruktur und zur Realisierung der Navigationsunterstützung mit Hilfe von Eigenschaftslinks.

7.1.1. INFORMATIONSCONTAINER ALS BAUSTEINE DER OBJEKTBEZOGENEN ORDNUNGSSTRUKTUR

- Anforderungen* Die objektbezogene Ordnungsstruktur soll das reale (oder geplante) baulich-räumliche Gefüge eines Bauwerks repräsentieren, wobei in erster Linie zentrale Einstiegspunkte zu den einzelnen fachspezifischen Repräsentationen eines Realweltobjektes bereitzustellen sind. Diese „Stellvertreter“ müssen jeweils die Identität und die Bedeutung des Realweltobjektes sowie strukturelle Beziehungen zwischen Realweltobjekten so beschreiben, dass auf Basis dieser Informationen Visualisierungen der objektbezogenen Ordnungsstruktur erzeugt werden können, die mit kognitiven Karten bzw. mentalen Modellen des betreffenden Bauwerks korrespondieren.

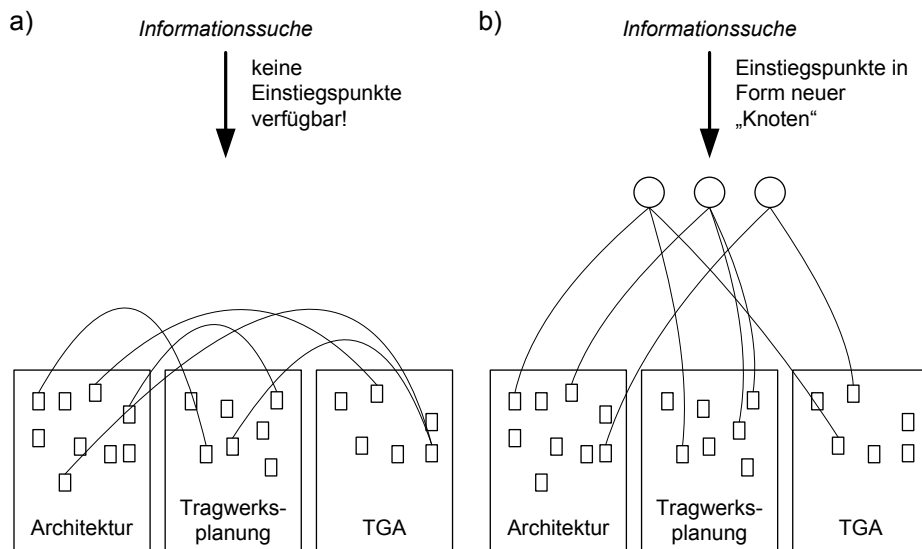


Abbildung 7.1.: Bereitstellung von Einstiegspunkten durch Knoten und Verknüpfungen

Die benötigten Einstiegspunkte können allein auf der Basis von Verknüpfungen zwischen einzelnen domänenspezifischen Repräsentationen, wie sie z. B. in [Willenbacher, 2002] vorgeschlagen werden, nicht realisiert werden (vgl. Abbildung 7.1 a). Erst durch eine Kombination von Verknüpfungen und neu hinzugefügten „Knoten“ sind Einstiegspunkte explizit verfügbar (siehe Abbildung 7.1 b).

Einstiegspunkte

Die Knoten stehen jeweils stellvertretend für je ein Realweltobjekt. Sie fassen Verweise auf die verschiedenen, dem Realweltobjekt entsprechenden fachspezifischen Repräsentationen zusammen. Bezug nehmend auf ihre Aufgabe der Zusammenfassung von Informationen werden sie im Folgenden als *Informationscontainer* bezeichnet.

Informationscontainer

Die Informationscontainer müssen jeweils eindeutig identifizierbar sein – sowohl für die verwaltende Software als auch für den Informationssuchenden. Aus der Sicht der Datenverwaltung ist diesbezüglich die Zuordnung

*Eindeutige
Identifikation*

eindeutiger Identifikatoren hinreichend. Aus der Sicht des Informationssuchenden ist die Identifizierbarkeit jedoch erst dann gegeben, wenn ein wahrgenommenes Objekt einer existierenden mentalen Repräsentation zugeordnet werden kann.

Wiedererkennbarkeit

In diesem Zusammenhang müssen die Eigenschaften Identität, Bedeutung und Struktur der Realweltobjekte durch die Informationscontainer abgebildet werden. Dafür sind folgende Informationen bereitzustellen:

- **Identität:** räumliche Dimensionen, Position und Informationen zur grafischen Darstellung
- **Bedeutung:** Kennzeichnung des Elementtyps
- **Struktur:** räumlich-topologische Beziehungen zu anderen Informationscontainern.

Struktur eines Informationscontainers

Jeder Informationscontainer benötigt bzw. enthält folglich:

- genau einen (persistenten) **eindeutigen Identifikator**, der die Identifikation des Containers durch das System bzw. Softwareapplikationen ermöglicht,
- eine beliebige Anzahl von Verweisen auf **domänenspezifische Repräsentationen**, d. h. auf Datenobjekte in den Domänenmodellen,
- einen oder mehrere Verweise auf einen **Elementtyp**, der die Bedeutung des vertretenen Realweltobjektes für Informationssuchende kennzeichnet,
- einen oder mehrere Verweise auf **Szeneobjekte**, die Informationen für die grafische Darstellung des Containers in 2D oder 3D-Sichten enthalten sowie
- beliebig viele **strukturelle Beziehungen** zu anderen Containern, die die baulich-räumliche bzw. topologische Struktur des Bauwerks beschreiben.

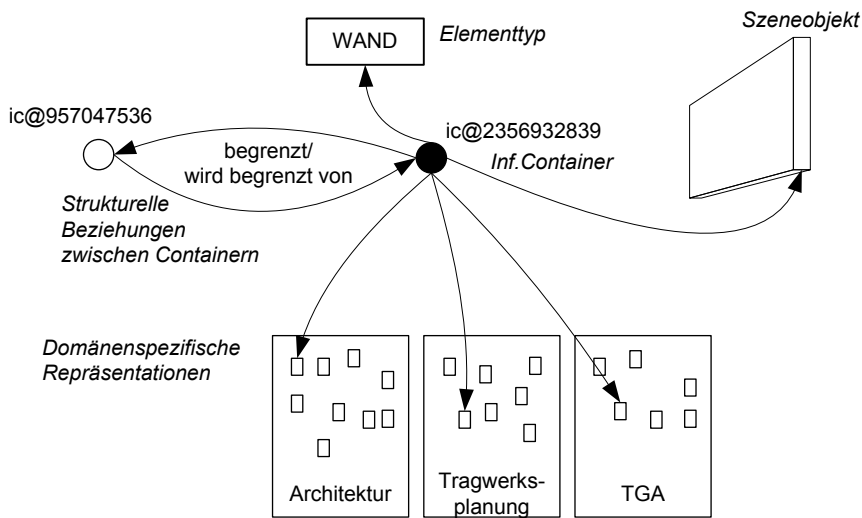


Abbildung 7.2.: Konkrete Ausprägung eines Informationscontainers für eine Wand (informale Darstellung)

Abbildung 7.2 zeigt exemplarisch die Struktur eines solchen Containers für ein Wandobjekt. Alle durch Verweise – dargestellt durch Pfeile in der Abbildung – referenzierten Informationen sind nicht Bestandteil der Erschließungsstruktur. Sie werden in Form externer Daten bzw. Inhalte verwaltet.

Trennung von Struktur und Inhalten

Für jeden der dargestellten Pfeile wird im Sinne des DataRef-Konzeptes von FOHM (vgl. Abschnitt 6.4.2) in der Erschließungsstruktur ein weiteres Knotenobjekt definiert, das einen Verweis auf ein externes Datenobjekt enthält. Dieses Knotenobjekt wird mit dem Informationscontainer verknüpft. Abbildung 7.3 zeigt die entsprechende Struktur des in Abbildung 7.2 dargestellten Informationscontainers in einer informalen Knoten-Kanten-Notation.

Referenzknoten

Die Verknüpfungen zwischen den Referenzknoten und den Informationscontainern werden dabei durch die Kanten-Objekte hergestellt. Auf diese Weise können Informationscontainer technisch ausschließlich durch einen

Identifikatoren als Kern der Informationscontainer

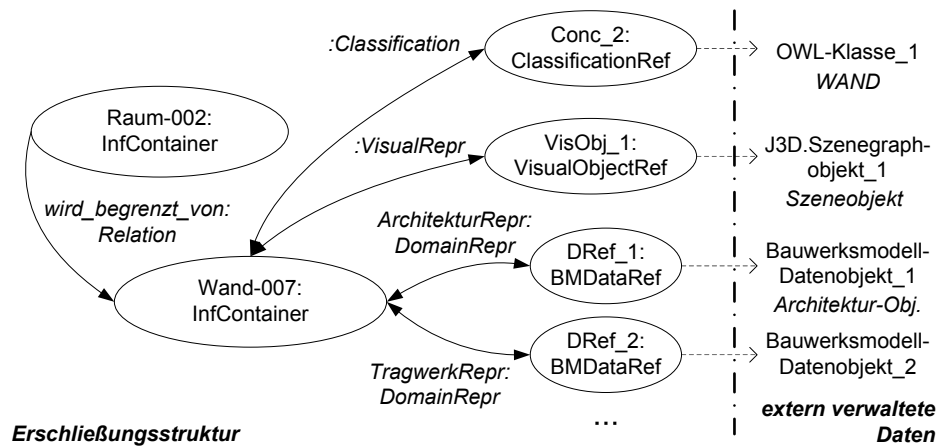


Abbildung 7.3.: Knoten-Kanten-Notation für ein Informationscontainer-Exemplar

eindeutigen Identifikator repräsentiert werden und müssen keinerlei Annahmen über die Anzahl und Struktur zugeordneter Verweise treffen.

Vorteile Dieser Ansatz hat zwei wesentliche Vorteile:

- Unabhängige Verwaltung externer Inhalte
Für die unterschiedlichen Arten zu referenzierender Informationen können jeweils verschiedene, der Art der Informationen angepasste Datenstrukturen und Verwaltungstechnologien verwendet werden.
- Erweiterbarkeit der Datenstruktur
Das Schema der Erschließungsstruktur kann um neue Typen von Referenzknoten erweitert werden, ohne dabei bereits bestehende Informationscontainer-Exemplare und deren Referenzen zu beeinflussen.

Bilden und Untersuchen von Teilmengen Während der Informationssuche muss auf der Basis der objektbezogenen Ordnungsstruktur bzw. der Informationscontainer das Bilden und Untersuchen von Teilmengen des Informationsraums möglich sein. Das Bilden von Teilmengen entspricht dabei der Auswahl einer Teilmenge der insge-

samt verfügbaren Informationscontainer. Wie diese Auswahl erfolgt, hängt jeweils von den durch die einzelnen Recherchemodule realisierten Interaktionstechniken ab. Das Untersuchen von Teilmengen erfolgt prinzipiell durch das Erkunden der mit den Informationscontainern verknüpften Informationen. Dafür müssen Recherchemodule diese Informationen entsprechend ihrer Präsentationsformen darstellen. Da grundsätzlich mehrere Recherchemodule zur Suche und Exploration verwendet werden können bzw. sollen, ist der Austausch der gebildeten Teilmengen zwischen den Modulen erforderlich – wobei jeweils Mengen von Informationscontainern auszutauschen sind.

Ein zum dargestellten Ansatz der Informationscontainer ähnliches Konzept wird in [Petzold u. a., 2003] beschrieben. Auch dort dienen Informationscontainer der Zusammenfassung von im Vorfeld nicht bekannten Informationen zu einem Objekt der Bauwerksstruktur. Im Unterschied zu dem in dieser Arbeit verfolgten Ansatz werden die Informationscontainer bei Petzold jedoch nicht ausschließlich zur Zusammenfassung, sondern gleichzeitig zur Verwaltung der Daten verwendet. Die erforderlichen Klassenstrukturen zur Verwaltung der Informationscontainer sind bei Petzold somit feingranularer und Änderungen derselben häufiger zu erwarten. Die in [Beer, 2006] beschriebenen *Elemente* und deren *Features* dienen ebenfalls einer von Fachapplikationen bzw. fachspezifischen Modellen unabhängigen und damit fachübergreifenden Repräsentation einzelner Bauwerkselemente. Sie werden in der zitierten Arbeit jedoch in erster Linie im Rahmen eines Versionierungsansatzes definiert und verwendet.

Parallelen zu anderen Ansätzen der Bauwerksmodellierung

Der Ansatz, alle ein Objekt beschreibenden Informationen auf einen eindeutigen Identifikator zu referenzieren, stellt ein Grundprinzip von Dokumentationssprachen dar (vgl. Abschnitt 6.4.1) und wird ebenso in computergestützten Nachschlagewerken angewendet. Diese Struktur ermöglicht die Unabhängigkeit der Referenzen von Namen oder Bezeichnungen ei-

Parallelen zu computergestützten Nachschlagewerken

nes Konzeptes. Einzelne Konzepte können auf diese Weise in mehreren Sprachen benannt und erläutert sowie um beliebige beschreibende Informationen ergänzt werden. Das International Framework for Dictionaries (IFD) beschreibt einen formalen Standard für entsprechende Datenstrukturen [Bjørkhaug u. Bell, 2007; ICIS, 2004].

Bauwesenspezifische Nachschlagewerke Auf der Basis von IFD entstanden in den vergangenen Jahren Pilotimplementierungen bauwesenspezifischer Nachschlagewerke [BARBi Project; Grant, 2008; Watson, 2008]. Diese konzentrieren sich jedoch auf die inhaltliche, teils mehrsprachige Beschreibung der in Bauwerksmodellen verwendeten Konzepte bzw. Klassen, nicht auf die Beschreibung der Realweltobjekte bzw. Elemente eines konkreten Bauwerks in einem fachübergreifenden Kontext.

7.1.2. UNTERSTÜTZUNG DER NAVIGATION INNERHALB KOMPLEXER MODELLSCHEMATA DURCH EIGENSCHAFTSLINKS

Problem komplexer Repräsentationen Wie zuvor beschrieben sollen Informationscontainer jeweils mit verschiedenen domänenspezifischen Repräsentationen der Realweltobjekte verknüpft werden. Die einzelnen domänenspezifischen Repräsentationen bestehen jedoch in der Regel nicht nur aus einem einzelnen Objekt, sondern aus einer Menge von Objekten unterschiedlicher Klassen, wobei diese Menge unter Umständen sehr komplexe Objektstrukturen aufweisen kann (vgl. Kapitel 3 sowie auch Abschnitt 6.3.3).

Beispiel Das Objektdiagramm in Abbildung 7.4 zeigt einen Ausschnitt der eine Wand repräsentierenden Objektmenge eines IFC-basierten Modells. Dieses am IFC-Schema orientierte Beispiel dient im Folgenden der Erläuterung des vorgeschlagenen Lösungsansatzes in seinen Grundprinzipien. Diese sind

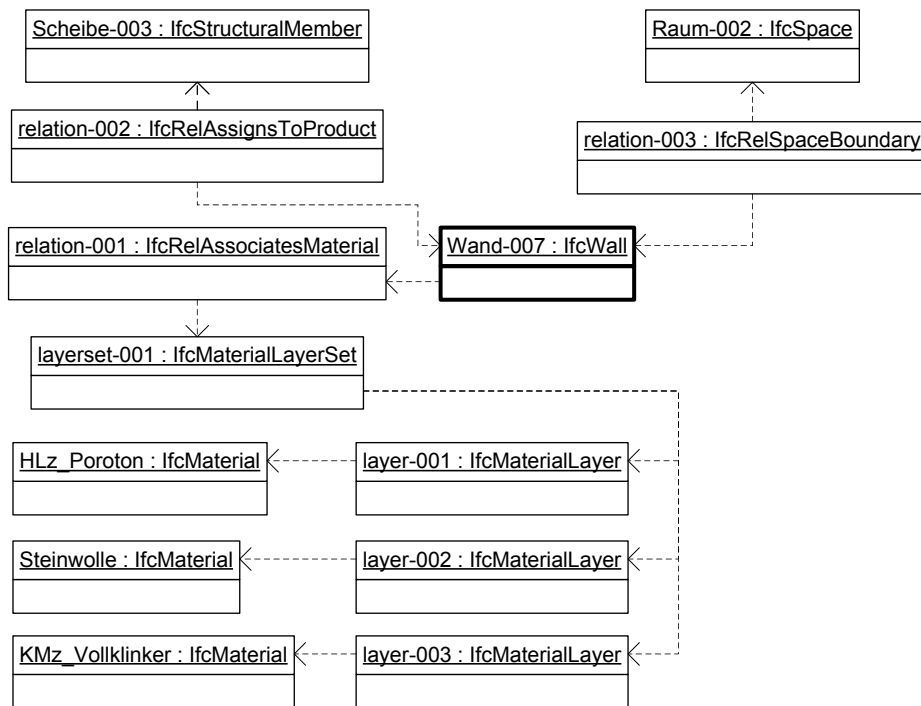


Abbildung 7.4.: Repräsentation einer Wand als Menge verknüpfter Objekte (UML-Objektdiagramm)

jedoch nicht an IFC gebunden, sondern auf jedes andere Modellschema übertragbar.

Referenzknoten beschreiben Einstiegspunkte Die Referenzknoten der Erschließungsstruktur verweisen jeweils auf das Objekt einer solchen Objektmenge, das das zentrale Konzept der domänenspezifischen Repräsentation bildet. Im Beispiel wäre dies das hervorgehobene Exemplar von `IfcWall` „Wand-007“. Damit wird jedoch nur der Einstieg in die entsprechende Objektmenge definiert.

Zweite Erschließungsebene Zur Unterstützung der Navigation innerhalb der domänenspezifischen Repräsentation ist die Einführung einer weiteren Erschließungsebene erforderlich. Diese muss Informationssuchende, die weder über Kenntnisse der tatsächlichen komplexen Objektstrukturen noch über Kenntnisse zu strukturierten Anfragesprachen verfügen, beim Auffinden relevanter Informationen für beispielsweise folgende Fragestellungen unterstützen:

- Aus welchen Materialien besteht „Wand-007“?
- Ist „Wand-005“ eine tragende Wand?

Beispiel: Abbildung von Eigenschaften in Domänenmodellen Aus der Sicht eines Entwerfenden sind dabei jeweils Eigenschaften einer Wand gesucht. Bauwerksmodelle bilden diese Eigenschaften jedoch oft durch eine Menge verknüpfter Objekte ab. Das Klassendiagramm in Abbildung 7.5 zeigt die an der Beschreibung „tragender Bauteile“ beteiligten Klassen im IFC-Schema.

Auswertung von Objektbeziehungen Die Mitwirkung eines Bauteils bei der Lastabtragung ist im IFC-Schema nicht als einfaches Attribut der Klasse `IfcBuildingElement` modelliert, sondern wird über eigenständige Tragwerkselemente (`IfcStructuralMember`) abgebildet. Exemplare beider Klassen sind jeweils über Relationen miteinander verknüpft. Das Ermitteln aller lastabtragenden Bauteile besteht somit im Ermitteln aller derjenigen Exemplare von `IfcBuildingElement`, die mit mindestens einem Exemplar von `IfcStructuralMember` verknüpft sind. Eine entsprechende formale Anfrage lautet in der Notation der Object

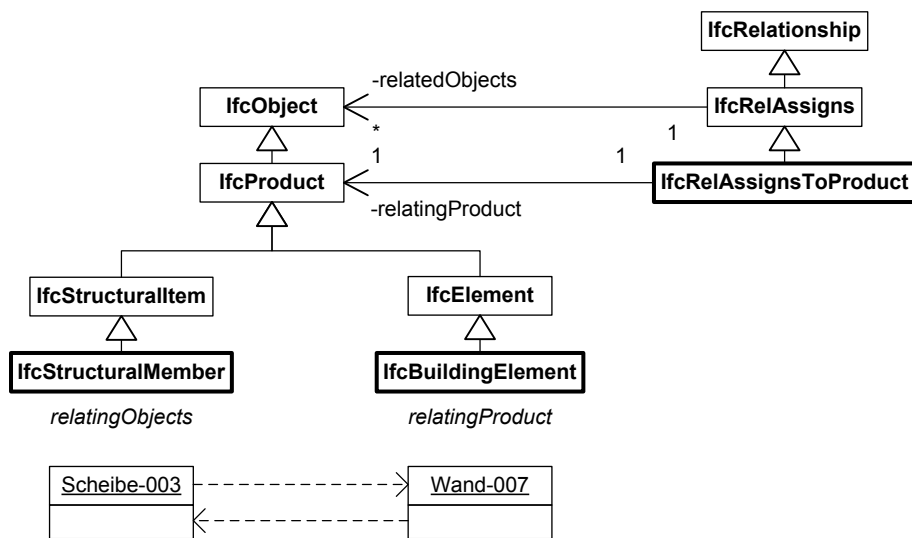


Abbildung 7.5.: IFC-Objektstruktur zur Verknüpfung eines Wandobjektes mit korrespondierenden Tragwerkselementen (UML-Klassendiagramm)

Query Language¹ [Elmasri u. Navathe, 2002] folgendermaßen (Listing 7.1):

```
1 SELECT STRUCT
   (buildingElement: bel, structuralMember: stm)
3 FROM  bel IN IfcBuildingElement,
       r IN bel.ReferencedBy,
5       stm IN r.RelatedObjects,
       stm IN IfcStructuralMember
7 WHERE stm = stm;
```

Listing 7.1: OQL-Anfrage zur Ermittlung aller Bauteile mit korrespondierenden Tragwerkselementen

```
8 SELECT STRUCT
   (buildingElement: bel, structuralMember: stm)
10 FROM bel IN IfcBuildingElement,
       r IN bel.ReferencedBy,
12       stm IN r.RelatedObjects,
       stm IN IfcStructuralMember
14 WHERE stm = stm AND bel.Name = "Wand-007";
```

Listing 7.2: OQL-Anfrage zur Ermittlung korrespondierender Tragwerkselemente zu einem einzelnen Bauteil

*Anfragemuster und
-ausprägungen*

Die Anfrage in Listing 7.1 definiert ausschließlich Restriktionen hinsichtlich der zu prüfenden Objektbeziehungen und beschreibt somit ein für alle Exemplare der Klasse `IfcBuildingElement` gültiges Anfragemuster zur Ermittlung jeweils korrespondierender Exemplare von `IfcStructuralMember`. Die Anfrage kann somit als ein auf Schema-Ebene gültiges Muster aufgefasst werden. Durch Hinzufügen einer weiteren Restriktion `bel.Name =`

¹ Zur Spezifikation siehe http://www.odbms.org/odmg_og.html

”Wand-007” in der *Where*-Klausel kann die Anfrage so modifiziert werden, dass ausschließlich die mit einem speziellen Exemplar von `IfcBuildingElement` korrespondierenden Exemplare von `IfcStructuralMember` ermittelt werden (siehe Listing 7.2). Dies kann als „Instanzieren” des allgemeinen Anfragemusters verstanden werden.

Mit Hilfe des Anfragemusters kann für die Klasse `IfcBuildingElement` eine neue Eigenschaft „korrespondierende Tragwerkselemente” definiert werden, die auf ein oder mehrere Exemplare von `IfcStructuralMember` verweist. Tatsächlich besitzt die Eigenschaft keine feste Wertebelegung, sondern die korrespondierenden Tragwerkselemente werden jeweils bei Bedarf für ein gegebenes Exemplar von `IfcBuildingElement` mit Hilfe des Anfragemusters ermittelt. Derartige Verknüpfungen werden im Folgenden als *Eigenschaftslink* bezeichnet. Zur Abbildung des beschriebenen Eigenschaftslinks muss die Erschließungsstruktur folgende Knoten- und Kantenobjekte enthalten (siehe Abbildung 7.6):

*Anfragemuster als
Klasseneigenschaften*

Eigenschaftslink

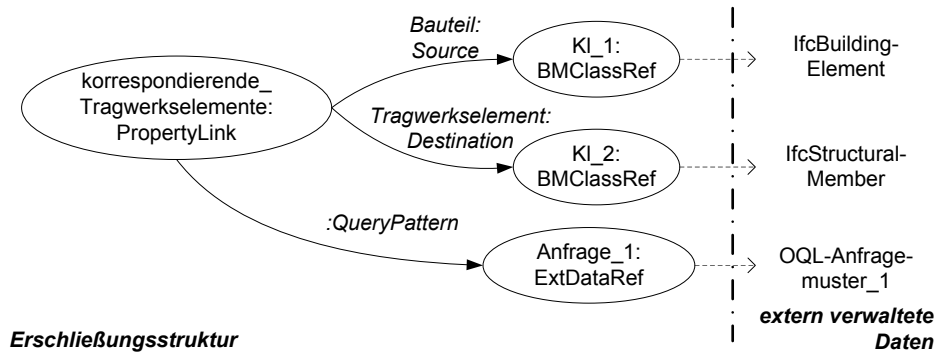


Abbildung 7.6.: Knoten-Kanten-Notation für ein Exemplar eines Eigenschaftslinks

Jedem Eigenschaftslink sind drei Referenzknoten zugeordnet: Einer der Knoten verweist auf das Konzept eines Domänenmodellschemas, für das mit Hilfe des Links eine zusätzliche Eigenschaft beschrieben wird. Ein wei-

*Struktur eines
Eigenschaftslinks*

terer verweist auf das Konzept, dessen Exemplare Werte der Eigenschaft darstellen. Ein dritter referenziert das entsprechende Anfragemuster. Das Anfragemuster selbst ist nicht Bestandteil der Erschließungsstruktur, sondern wird als extern zu verwaltender Inhalt betrachtet.

Auswertung zum Zeitpunkt der Anfrage Diese Eigenschaftslinks können zum Zeitpunkt der Anzeige von Informationscontainern oder Domänenmodelldaten durch das System verwendet werden, um automatisch verfügbare Domänenmodelldaten auszuwerten, zusammenzufassen und zu präsentieren. Dafür muss das System jeweils anhand einzelner Referenzknoten bzw. Datenobjekte prüfen, ob für sie Eigenschaftslinks in der Erschließungsstruktur definiert wurden, die entsprechenden Anfragen instanziiieren und auswerten sowie entsprechende Präsentationen erzeugen.

Vorteile der Eigenschaftslinks Das Konzept der Eigenschaftslinks bietet gegenüber Ansätzen, bei denen Informationen zur Auswertung und Präsentation komplexer Objektstrukturen fest in den Viewer-Werkzeugen kodiert sind, zwei wesentliche Vorteile:

- Einmal erstellte Eigenschaftslinks bzw. deren Anfragemuster können durch unterschiedliche Recherchemodule für die Präsentation komplexer Objektstrukturen verwendet werden.
- Weitere Eigenschaftslinks können definiert und durch Viewer verwendet werden, ohne dass ein Rekompilieren bestehender Viewer bzw. Recherchemodule erforderlich ist.

Parallelen in OHS-Ansätzen Ein ähnlicher Ansatz wurde in [Weal u. a., 2001] beschrieben. Mittelpunkt dieser Forschungsarbeiten ist ein verteiltes Informationsmanagementsystem, das mit Hilfe von Softwareagenten und Ontologien Daten aus verschiedenen Quellen zu einem Informationsraum integriert. Gegenstand dieses Informationsraums sind Mitarbeiter, Objekte und Aktivitäten in einem Forschungslabor. Um Informationssuchenden die Interaktion mit diesem Informationsraum zu erleichtern, wurden Links zwischen Ontologie-Konzepten

und Anfragen definiert und in einem separaten Hypermedia-Modell auf der Basis von FOHM (siehe Abschnitt 6.4.2) verwaltet. Die Anfragen repräsentieren dabei eine von Experten für die Nutzer des Systems vorkonfigurierte Sicht auf den Informationsraum. Informationssuchende interagieren ausschließlich über einen Webbrowser mit dem Informationsraum. Serverseitig werden dafür anhand der im Informationsmanagementsystem verfügbaren Daten und der vom Hypermedia-Modell bereitgestellten Anfrage-Links dynamisch HTML-Dokumente generiert, die an den Webbrowser übermittelt werden. Sobald Benutzer die im HTML-Dokument enthaltenen, auf eine Anfrage verweisenden Links aktivieren, wertet das System im Hintergrund die entsprechende Anfrage aus und präsentiert die Ergebnisse in einem weiteren dynamisch generierten HTML-Dokument.

Das vorgeschlagene Konzept der Eigenschaftslinks lehnt sich in folgenden Aspekten an den oben skizzierten Ansatz an:

- Beschreibung auszuwertender Objektbeziehungen in Form von Anfragemustern und Verknüpfung der Anfragemuster mit einzelnen Klassen der Domänenmodelle
- Separate Verwaltung dieser Links und Anfragemuster als Teil der Erschließungsstruktur (getrennt vom Bauwerksmodellverbund)
- Ausprägung und Auswertung der Anfragemuster korrespondierend zu Benutzerinteraktionen
- Zusammenführung von Bauwerksmodellldaten, Anfragemustern und Anfrageergebnissen zum Zeitpunkt der Präsentation für den Nutzer
- Erstellen der Anfragemuster und Links durch Domänenexperten im Vorfeld der Systemnutzung.

Unterschiede Der in dieser Arbeit vorgeschlagene Ansatz geht jedoch von einer universelleren Verwendbarkeit der Eigenschaftslinks aus, als in [Weal u. a., 2001] beschrieben. Das serverseitige Generieren von Präsentationsdokumenten ist für die angestrebte modular aufgebaute Nutzerschnittstelle nicht sinnvoll, da die einzelnen Module jeweils ihre eigenen Präsentationstechniken realisieren sollen und bezüglich dieser vorab keine Annahmen getroffen werden können. Stattdessen ist es Aufgabe der einzelnen Module, entsprechend ihrer Intention die verfügbaren Eigenschaftslinks zur Auswertung, Aufbereitung und Präsentation von Domänenmodellldaten zu verwenden.

7.2. KONZEPTUELLES SCHEMA DER ERSCHLIESSUNGSSTRUKTUR

Ausgehend von den erläuterten Lösungsansätzen wird in den folgenden Abschnitten ein objektorientiertes konzeptuelles Schema entwickelt, das als Datenstruktur zur Abbildung und Verwaltung der Erschließungsstruktur dient.

7.2.1. RELEVANTE KONZEPTE

Knoten- und Kantentypen Zur Repräsentation von Informationscontainern und Eigenschaftslinks in einer objektorientierten Datenstruktur werden die in Abbildung 7.7 zunächst informal dargestellten Typen von Knoten- und Kantenelementen benötigt. Knotentypen repräsentieren prinzipiell atomare Informationseinheiten. Kantentypen stehen für verschiedene Arten von Beziehungen zwischen den Informationseinheiten.

Das Klassendiagramm in Abbildung 7.8 zeigt die entsprechend benötigten Klassen zur Abbildung der Erschließungsstruktur.

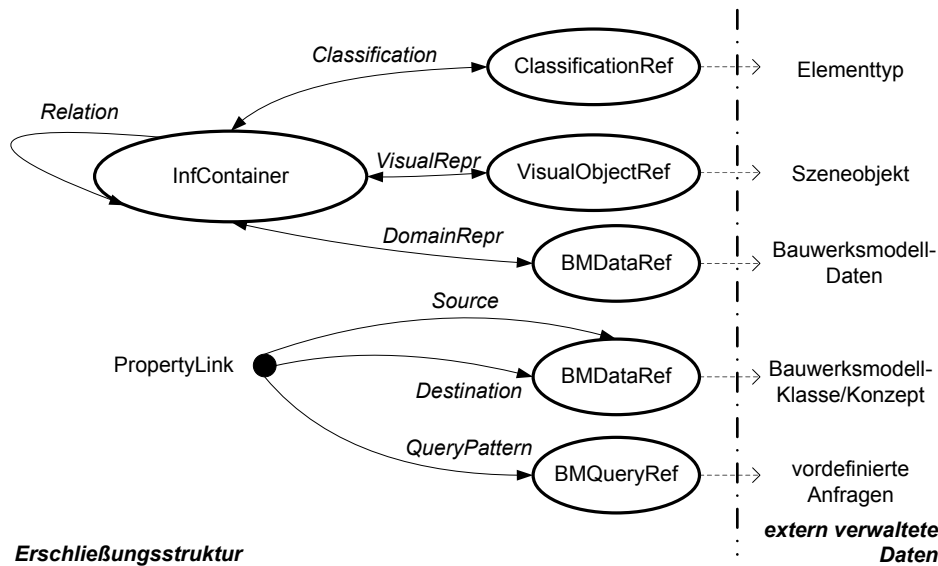


Abbildung 7.7.: Knoten- und Kantenarten zur Abbildung der Erschließungsstruktur

Infcontainer-Exemplare vertreten jeweils ein Realweltobjekt. Sie bilden in ihrer Gesamtheit die objektbezogene Ordnungsstruktur als erste Ebene der Erschließungsstruktur. PropertyLink-Exemplare repräsentieren Eigenschaftslinks. Sie bilden die zweite Ebene der Erschließungsstruktur. Daneben dienen vier verschiedene Typen von Referenzknoten der Verwaltung von Verweisen auf externe Inhalte bzw. Daten.

Ebenen der Erschließungsstruktur

Auf der Basis der referenzierten Informationen soll durch die Recherche-module u. a. die Darstellung der objektbezogenen Ordnungsstruktur realisiert werden. Dabei sind grundsätzlich verschiedene Prinzipien der Darstellung möglich, die von der Art der durch einen Knotentyp referenzierten Informationen abhängen. Tabelle 7.1 gibt einen Überblick über mögliche Darstellungsprinzipien in Korrespondenz zu den einzelnen Knotentypen.

Mit Hilfe der Assoziation Classification und einem Knotenobjekt vom Typ ClassificationRef kann einem Informationscontainer ein Element-

Semantik von Informationscontainern

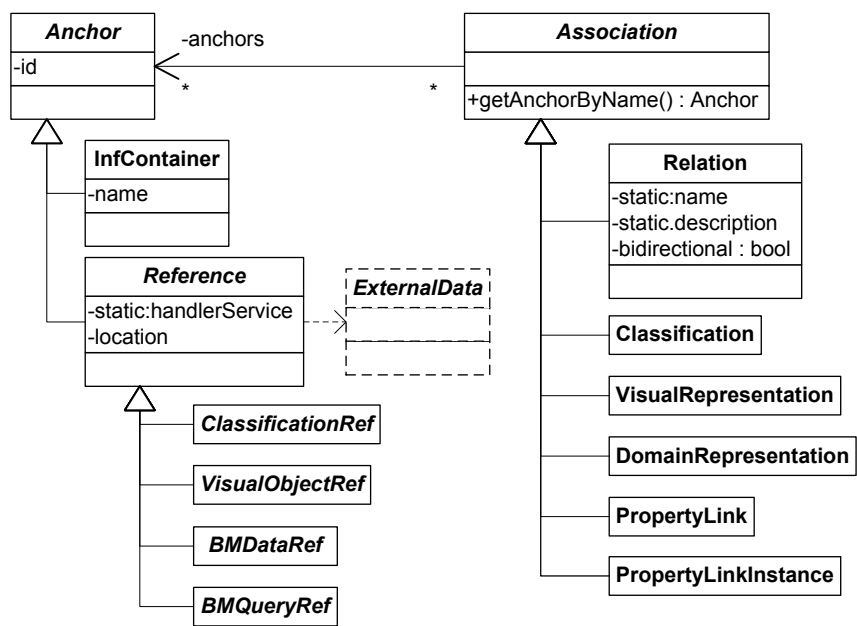


Abbildung 7.8.: Klassen des ES-Schemas (UML-Klassendiagramm)

Klasse	Anwendbares Darstellungsprinzip
Relation (und Unterklassen)	Allgemeine Beziehungsstrukturen innerhalb des Gebäudes, Aggregationsstruktur des Gebäudes, Beziehungen eines einzelnen <code>InfContainer</code> -Exemplars zu anderen Exemplaren
VisualRepresentation/ VisualObjectRef	Interaktive 3D-Szene des Gebäudes, 2D-Plandarstellungen (Grundrisse, Schnitte)
Classification/ ClassificationRef	Sortierung/Liste nach Elementtypen
DomainRepresentation/ BMDataRef	Sortierung/Liste nach verfügbaren Domänenmodellen, Übersicht zu einem einzelnen <code>InfContainer</code> -Exemplar: verfügbare Repräsentationen in den Domänenmodellen
PropertyLink/ PropertyLinkInstance	Sortierung/Liste nach in den Domänenmodellen beschriebenen Eigenschaften, Übersicht zu einem einzelnen <code>InfContainer</code> -Exemplar: Eigenschaften des Realweltobjektes in einzelnen domänenspezifischen Repräsentationen

Tabelle 7.1.: Mögliche Darstellungsprinzipien für die Knoten- und Assoziationsklassen

typ zugeordnet werden, der die Bedeutung oder Semantik des Informationscontainers – in erster Linie für den Informationssuchenden – beschreibt. Die Menge der jeweils gültigen Elementtypen muss dabei in einem extern beschriebenen und verwalteten Vokabular oder Klassifikationssystem definiert werden. Dieses wird durch `ClassificationRef`-Exemplare referenziert.

Auf diese Weise ist die Zuordnung von Semantik zu Informationscontainern variabel und kann jederzeit geändert werden, ohne die übrigen mit einem Container verknüpften Informationen zu beeinflussen.

Variable Zuordnung

Knotenobjekte vom Typ `VisualObjectRef` referenzieren Visualisierungsinformationen, die die Identität eines Informationscontainers beschreiben.

Identität von Informationscontainern

Die Zuordnung zu `InfContainer`-Exemplaren erfolgt über `VisualRepresentation`-Assoziationen.

Zur Visualisierung eines Informationscontainers in einer interaktiven 3D-Szene sind jeweils folgende Informationen erforderlich:

- die Geometrie bzw. räumliche Dimension des Realweltobjektes, beispielsweise als Beschreibung der begrenzten Oberflächen,
- grafische Attribute zur Darstellung der Geometrie sowie
- die Position der Geometrie in einem Bezugskordinatensystem.

Szeneobjekte Diese Informationen werden im Folgenden unter dem Begriff *Szeneobjekt* zusammengefasst. Es wird davon ausgegangen, dass jedem Informationscontainer mindestens ein solches Szeneobjekt zugeordnet ist. Die vorgeschlagene Datenstruktur erlaubt jedoch die Zuordnung von mehr als einem Szeneobjekt. Weitere Szeneobjekte können so beispielsweise Informationen für zweidimensionale Plandarstellungen bereitstellen.

Verweise auf domänenspezifische Repräsentationen Knotenobjekte vom Typ `BMDataRef` verwalten Zeiger auf Bauwerksmodelldaten. Zusammen mit `DomainRepresentation`-Assoziationen definieren sie Verweise von Informationscontainern auf domänenspezifische Repräsentationen des betreffenden Realweltobjekts.

Aufbau von Eigenschaftslinks Der Knotentyp `BMClassRef` referenziert einzelne Klassen von Domänenmodellschemata und ist Bestandteil der Definition von Eigenschaftslinks. Ein weiterer Knotentyp `ExtDataRef` verwaltet Zeiger auf extern verwaltete Anfragemuster als Auswertungsvorschrift von Eigenschaftslinks.

Struktur – Beziehungen zwischen Containern Strukturelle Beziehungen zwischen Realweltobjekten bzw. Informationscontainern als deren „Stellvertreter“ werden über Kanten vom Typ `Relation` definiert.

7.2.2. KLASSENHIERARCHIE

`Anchor` bildet die abstrakte Oberklasse für alle Knotenklassen. Das von `Anchor` definierte Attribut `id` dient der Zuordnung eindeutiger Identifikatoren zu jedem Knotenobjekt. `InfContainer` spezialisiert `Anchor` und fügt ein weiteres Attribut `name` hinzu. Dieses verwaltet einen für Menschen lesbaren Bezeichner eines Informationscontainer-Exemplars, beispielsweise „Wand-007“.

Klasse `Anchor`

`AbstrReference` definiert die Oberklasse für alle Knotenobjekte, die Verweise auf extern verwaltete Daten enthalten. Das Attribut `location` dient dabei der Spezifikation des Speicherorts, `handlerService` enthält einen Bezeichner für einen Dienst, der die zur Auswertung und Verarbeitung der Referenz benötigte Applikationslogik implementiert. Da beide Eigenschaften unmittelbar von der im konkreten Fall verwendeten Technologie der Datenverwaltung abhängig sind, bleibt die weitere Spezifikation Aufgabe der konkreten Unterklassen von `AbstrReference`. Diese Struktur erlaubt ein unproblematisches Einbinden neuer Inhalte (Arten externer Daten) durch Ableiten weiterer konkreter Knotenklassen zur Referenzierung dieser Inhalte.

Klasse

`AbstrReference`

Korrespondierend zu `Anchor` ist `Association` die abstrakte Oberklasse aller Verknüpfungsklassen. Dabei sind Verknüpfungen nicht nur auf gerichtete Beziehungen (Kanten) zwischen jeweils genau zwei Knotenobjekten beschränkt, sondern können jeweils eine beliebige Anzahl von Knoten referenzieren. Die Knotenreferenzen werden als geordnete Bezeichner-Wert-Paare verwaltet, so dass anhand eines Bezeichners, genau ein zugehöriges Knotenobjekt ermittelt werden kann. Dafür muss jede Unterklasse von `Association` die für ihre Exemplare gültige Anzahl zu referenzierender Knotenobjekte sowie deren Bezeichner festlegen.

Klasse `Association`

Eigenschaftslinks = Assoziationen = Eigenschaftslinks werden nach diesem Prinzip als dreistellige Assoziationen abgebildet, die jeweils Referenzen auf genau eine Quellklasse, eine Zielklasse und ein Anfragemuster besitzen. Mit Hilfe des Anfragemusters können dabei jeweils verknüpfte Exemplare der Zielklasse zu gegebenen Exemplaren der Quellklasse ermittelt werden.

Informationscontainer = Knoten = Informationscontainer müssen als Knoten abgebildet werden, da ihnen jeweils eine individuell unterschiedliche Anzahl von Referenzen auf externe Inhalte zugeordnet werden soll. Während der Lebensdauer eines Informationscontainer-Exemplars kann sich die Anzahl der Referenzen zudem ändern.

7.2.3. STRUKTURELLE BEZIEHUNGEN ZWISCHEN INFORMATIONSCONTAINERN

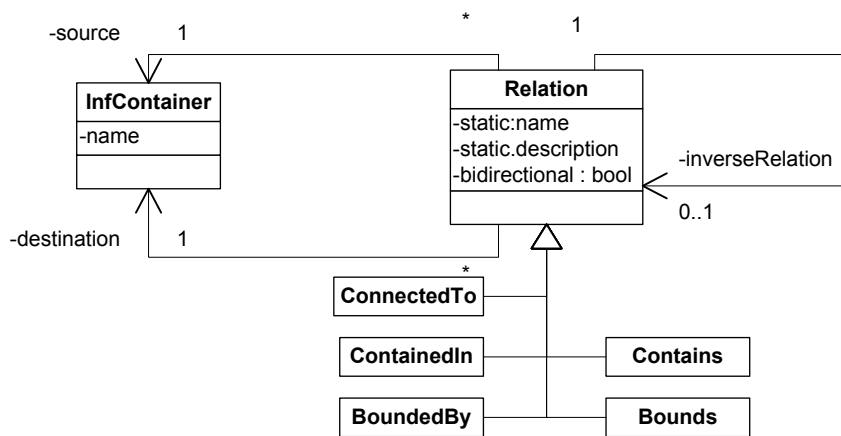


Abbildung 7.9.: Abbildung von Beziehungen zwischen Informationscontainern (UML-Klassendiagramm)

Klasse Relation = Die Klasse **Relation** bildet die Oberklasse einer erweiterbaren Hierarchie strukturbeschreibender Assoziationsklassen. Wie im Klassendiagramm

in Abbildung 7.9 dargestellt, definiert `Relation` genau zwei Verweise auf Knotenobjekte vom Typ `InfContainer`. Unter dem Bezeichner `source` wird auf das `InfContainer`-Exemplar verwiesen, von dem die Beziehung ausgeht (das Subjekt). Der Bezeichner `destination` kennzeichnet das `InfContainer`-Exemplar, das das Objekt der Beziehung repräsentiert.

`Relation` definiert zwei statische Klassenattribute `name` und `description` für einen Bezeichner und eine kurze Erläuterung. Beides dient der Erläuterung der Semantik des jeweiligen Beziehungstyps für Informationssuchende, d. h. der zugewiesene Wert stellt eine Eigenschaft der Klasse dar, und ist für ihre Exemplare nicht variabel. Spezialisierende Unterklassen von `Relation` können jeweils die Wertebelegung dieser Attribute ändern. Darüber hinaus enthält `Relation` eine auf sich selbst verweisende Assoziationsbeziehung mit der Rollenbezeichnung `inverseRelation`. Diese erlaubt in Kombination mit dem Attribut `bidirectional` die Modellierung bidirektionaler Beziehungen bzw. unidirektionaler Beziehungen mit Kennzeichnung des zugehörigen inversen Beziehungsexemplars.

*Semantik des
Beziehungstyps*

Einige häufig benötigte Beziehungstypen werden als bereits definierte Unterklassen von `Relation` im Sinne eines Basisschemas für typische Anwendungsfälle vorgegeben. Diese sind Tabelle 7.2 zu entnehmen.

*Unterklassen von
Relation*

Mit Hilfe dieser Unterklassen können beispielsweise folgende Beziehungen abgebildet werden:

- „Raum-012“ befindet sich im „Erdgeschoss“
- „Stütze-005“ befindet sich in „Raum 008“
- „Wand-007“ begrenzt „Raum-008“.

Unterklasse Relation	von Bedeutung
ConnectedTo	Beschreibt Verbindungen zwischen Bauteilen oder Räumen (bidirektional)
ContainedIn / Contains	Beschreiben Aggregationsbeziehungen zwischen Bauteilen bzw. Räumen (jeweils unidirektional)
BoundedBy / Bounds	Beschreiben Begrenzungen von Räumen durch Bauteile (jeweils unidirektional)

Tabelle 7.2.: Beziehungsklassen und deren Bedeutung

7.2.4. ZUORDNUNG DOMÄNENSPEZIFISCHER REPRÄSENTATIONEN

Klasse Domain-Representation

Für die Zuordnung domänenspezifischer Repräsentationen zu Informationscontainern sind Exemplare der Assoziationsklasse **DomainRepresentation** verantwortlich. Die daran beteiligten Klassen zeigt Abbildung 7.10.

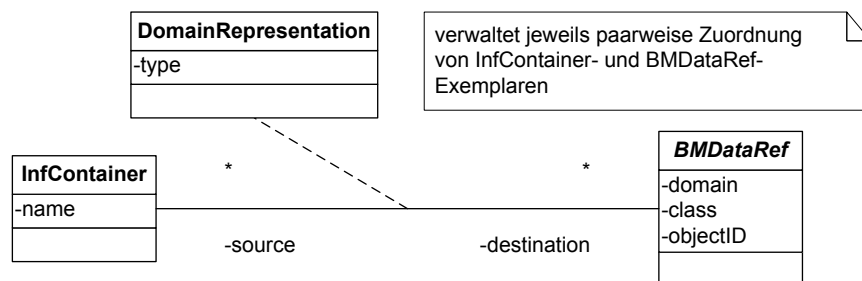


Abbildung 7.10.: Zuordnung domänenspezifischer Repräsentationen (UML-Klassendiagramm)

Klasse BMDataRef

Die Klasse **BMDataRef** definiert den Knotentyp zur Referenzierung extern verwalteter Bauwerksmodellldaten. Die drei Attribute von **BMDataRef**

enthalten jeweils den Identifikator des referenzierten Datenobjektes, den Namen des Domänenmodells, dem das Datenobjekt angehört, sowie den Name seiner Klasse im Domänenmodellschema.

Jedem Informationscontainer kann eine beliebige Anzahl von Repräsentationen in verschiedenen Domänenmodellen zugeordnet sein. Dabei sind grundsätzlich vier verschiedene Fälle logischer Zuordnung zu unterscheiden:

Logische Zuordnung

- Entsprechung
- Zusammenfassung
- Zerlegung
- attributive Eigenschaft.

Entsprechung. Ein Datenobjekt entspricht einem Informationscontainer, wenn sich beide auf ein und dasselbe Realweltobjekt beziehen. „Fenster_S1“ in der objektbezogenen Ordnungsstruktur entspricht „Fenster-012“ im Architekturmodell.

Zusammenfassung. Ein Informationscontainer kann mehrere Datenobjekte ein und desselben Domänenmodells zusammenfassen. „Wand_S1“ in der Ordnungsstruktur wurde im Architekturmodell durch die Wandabschnitte „Wand-003“ bis „Wand-008“ modelliert.

Zerlegung. Ein Datenobjekt kann in der Ordnungsstruktur durch mehrere Informationscontainer repräsentiert sein. Der „Aufzugsschacht_K2“, im Tragwerksmodell als ein aussteifendes Element modelliert, wird in der Ordnungsstruktur durch 4 einzelne Wände repräsentiert.

Attributive Eigenschaft. Datenobjekte können einzelne Eigenschaften eines Realweltobjektes beschreiben. Die Dokumentation eines Risses beschreibt den Zustand der „Wand_W2“.

Zur Kennzeichnung des Zuordnungstyps definiert die Klasse `DomainRepresentation` ein Attribut `type`, das einen der Werte aus Tabelle 7.3 annehmen kann.

Wert von <code>type</code>	Bedeutung
EQUAL	Entsprechung
IS_PART_OF	Zusammenfassung
CONSISTS_OF	Zerlegung
ATTRIBUTE	attributive Eigenschaft

Tabelle 7.3.: Zulässige Werte des `type`-Attributs zur Kennzeichnung der logischen Zuordnung zwischen Informationscontainern und domänenspezifischen Repräsentationen

7.2.5. ZUORDNUNG VON SZENEOBJEKTEN UND KLASSIFIKATIONSSYSTEMEN

Klassen `VisualObjectRef` und `ClassificationRef` definieren Zeiger auf extern verwaltete Szeneobjekte bzw. Elementtypen in einem externen Vokabular oder Klassifikationssystem. Assoziationen vom Typ `VisualRepresentation` bzw. `Classification` verknüpfen die Knoten-Exemplare jeweils mit einem Informationscontainer (siehe Klassendiagramme der Abbildungen 7.11 und 7.12).

7.2.6. EIGENSCHAFTSLINKS

Klasse `PropertyLink` Eigenschaftslinks werden durch die Assoziationsklasse `PropertyLink` sowie die beiden Knotenklassen `BMDataRef` und `BMQueryRef` modelliert, wie Abbildung 7.13 zeigt.

7.2. KONZEPTUELLES SCHEMA DER ERSCHLIESSUNGSSTRUKTUR

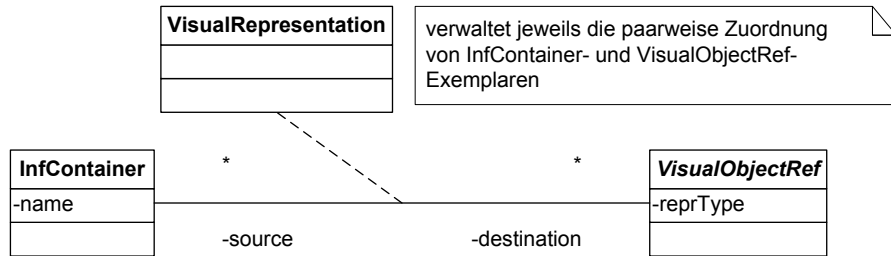


Abbildung 7.11.: Zuordnung von Szeneobjekten (UML-Klassendiagramm)

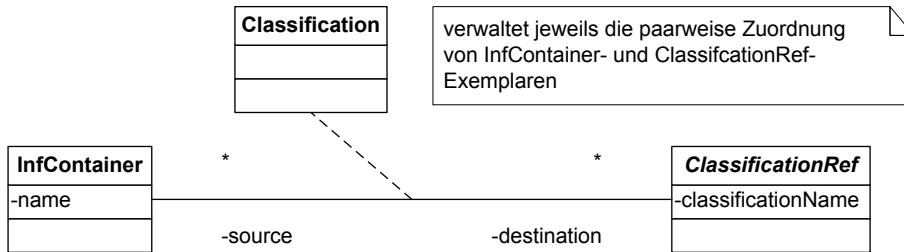


Abbildung 7.12.: Zuordnung von Typen aus externen Klassifikationen (UML-Klassendiagramm)

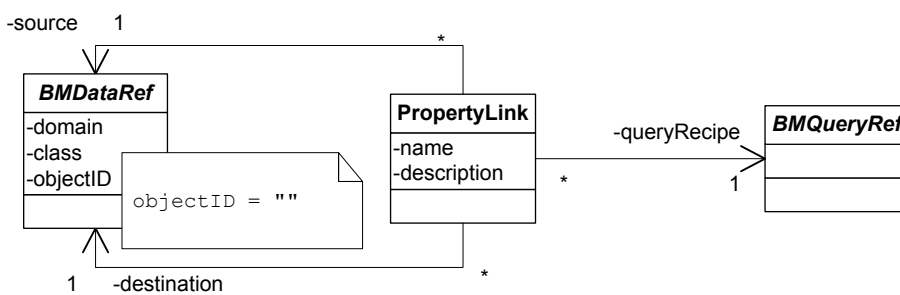


Abbildung 7.13.: Eigenschaftslinks (UML-Klassendiagramm)

PropertyLink-Exemplare repräsentieren dabei eine dreistellige, d. h. eine über drei Referenzknoten verfügende Assoziation. Ein Knoten vom Typ BMDataRef zeigt dabei auf die Klasse eines Domänenmodellschemas, der die Eigenschaft zugeordnet ist, die der Link beschreibt. Ein weiteres Exemplar von BMDataRef zeigt auf die Klasse, deren Exemplare gültige Werte der Eigenschaft darstellen. Diese BMDataRef-Exemplare verwalten dafür jeweils den Namen des betreffenden Domänenmodells sowie den Namen der referenzierten Klasse. Das Attribut objectID bleibt „leer“, da die vom Link beschriebene Eigenschaft für alle Objekte der betreffenden Klasse gültig ist.

Klasse BMQueryRef

Über Knoten vom Typ BMQueryRef referenzieren Eigenschaftslinks extern repräsentierte Anfragemuster als Vorschrift zur Ermittlung verknüpfter Exemplare der Quell- und Zielklasse. Die Klasse PropertyLink definiert darüber hinaus zwei Attribute zur Verwaltung eines Bezeichners und einer inhaltlichen Erläuterung der Eigenschaft. Im Gegensatz zu Relation sind diese beiden Attribute jedoch keine statischen Klassenattribute. Die Werte beschreiben jeweils die spezifische Semantik des jeweiligen Exemplars und werden somit für jedes Exemplar individuell gesetzt.

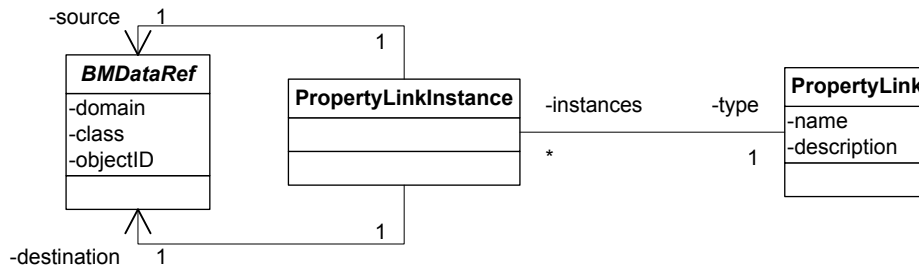


Abbildung 7.14.: Temporäre Ausprägung von Eigenschaftslinks für Modell-
daten (UML-Klassendiagramm)

*Temporäre
Ausprägung auf
Datenebene*

Wird ein Eigenschaftslink während einer Recherchesitzung auf Anforderung des Informationssuchenden ausgewertet, sollen die Ergebnisse tempo-

rär für die Rechercsesitzung vorgehalten werden. Dafür werden auf Datenebene temporäre Verknüpfungen angelegt, die konkrete Ausprägungen eines Eigenschaftslinks repräsentieren, indem sie zwei konkrete Datenobjekte miteinander verknüpfen. Die entsprechenden Klassen zeigt Abbildung 7.14

Ein `PropertyLinkInstance`-Exemplar verweist dabei jeweils auf zwei Exemplare von `BMDataRef` als Quelle und Ziel des Links sowie auf ein `PropertyLink`-Exemplar das den zugehörigen Linktyp repräsentiert.

7.3. FORMALE REPRÄSENTATION

Zur Repräsentation und Verwaltung der Erschließungsstruktur wurden im Rahmen der Arbeit exemplarisch zwei Ansätze untersucht.

Mögliche Ansätze...

Unter Verwendung eines Dynamik unterstützenden Modellverwaltungssystems (MVS) [Willenbacher, 2002; Hauschild, 2003] kann die Erschließungsstruktur als ein eigenständiges Modell innerhalb des variablen Modellverbundes behandelt und verwaltet werden. Dieser Ansatz wurde im Prototypen *Navi Model Editor* verfolgt (siehe Anhang D.3). Die verfügbaren Funktionalitäten des als Prototyp bestehenden MVS Servers sowie einiger Client-Anwendungen sind jedoch in erster Linie auf die Verwaltung der Modelle ausgerichtet, Funktionalitäten zur Erstellung und Bearbeitung von Schemata und Daten sind nur rudimentär verfügbar.

*...als separates Modell
im variablen Verbund*

Einen anderen Weg stellt die Repräsentation der Erschließungsstruktur, d.h. des konzeptuellen Schemas und der Daten unter Verwendung einer formalen Beschreibungssprache dar, für die sehr komfortable Editoren frei verfügbar sind. Im Rahmen eines weiteren Prototypen kam dabei die Web Ontology Language zum Einsatz (siehe Anhang D.1 sowie Anhang ??). Dies stellt keinen grundsätzlichen Widerspruch zum ersten Ansatz dar, da auf der

*...mit Hilfe einer
formalen
Beschreibungssprache*

Basis einer formalen Schema- und Datenbeschreibung ein entsprechendes Modell im MVS prinzipiell automatisiert erzeugt werden kann.

7.3.1. ONTOLOGIEN UND SEMANTIC WEB TECHNOLOGIEN

Der Ontologie-Begriff

Der Begriff *Ontologie* stammt ursprünglich aus der Philosophie und bezeichnet dort die Lehre vom Sein und den Bedingungen des menschlichen Seins [Meixner, 2004]. In der Informatik steht der Begriff Ontologie für ein Modell eines Anwendungsbereiches, das zur Kommunikation zwischen verschiedenen Benutzern verwendet wird [Leser u. Naumann, 2007]. In diesem Sinne definierte Gruber den Begriff Ontologie als „*formale Beschreibung der Konzepte und Beziehungen, die für einen [Software-]Agenten oder eine Gemeinschaft von Agenten existieren können*“ [Gruber, 1993]. Allgemeiner formuliert ist eine Ontologie in der Informatik also eine formale Konzeption, die es Softwareapplikationen bzw. Softwareagenten erlaubt, die Bedeutung von Daten auszutauschen bzw. gemeinsam zu verwenden. Dies setzt eine maschinenlesbare Repräsentation der Ontologie voraus, wofür sich formale Sprachen zur Wissensrepräsentation auf der Basis der *Beschreibungslogik* eignen [Leser u. Naumann, 2007]. Diese Art der Repräsentation erlaubt das maschinengestützte Ableiten neuer Informationen aus den in der Ontologie beschriebenen Informationen durch logisches Schlussfolgern (Inferenzverfahren).

Semantic Web Technologien

In den letzten Jahren gewannen Ontologien im Zusammenhang mit der Vision des Semantic Web an Interesse. Unter dem Begriff Semantic Web wird eine Weiterentwicklung des World Wide Web (WWW) verstanden, bei der durch Annotation die Inhalte von Webseiten für Maschinen bzw. Software lesbar und interpretierbar gemacht werden. Ziel ist dabei, die Bedeutung der Inhalte für Maschinen explizit zu machen, um so das computergestützte Management der im WWW verfügbaren Informationen zu verbessern

[Leser u. Naumann, 2007]. Im Zuge dieser Bemühungen erfuhren verschiedene Softwaretechnologien und informationstechnische Konzepte einen Aufschwung bzw. eine Weiterentwicklung, darunter vor allem Techniken und Sprachen zur formalen Wissensrepräsentation sowie Technologien zur Realisierung flexibler webbasierter Softwareapplikationen, wie Softwareagenten, Web Services und entsprechende Architekturkonzepte. Gegenwärtig stellt die Web Ontology Language (OWL) den vom World Wide Web Consortium (W3C) offiziell empfohlenen Sprachstandard zur formalen, expliziten Repräsentation von Ontologien für das Semantic Web dar².

Technologien des Semantic Web wurden von verschiedenen Forschungsaktivitäten im Bauwesen aufgegriffen. So werden Ontologien beispielsweise als vermittelnde Schicht zwischen Ingenieuren als „End-Benutzer“ auf der einen Seite und umfangreichen Bauwerksinformationsmodellen, wie z. B. IFC, auf der anderen Seite eingesetzt [Gehre u. a., 2006]. Ein ähnlicher Ansatz ist bereits in [Katranuschkov u. a., 2003] beschrieben. Im Rahmen der zitierten Arbeit wurde ein Ontologie-Framework entwickelt, das die Integration und Interoperabilität in *virtuellen Organisationen* auf allgemeiner Ebene ermöglichen soll. Andere Arbeiten fokussieren stärker auf die automatisierte Kommunikation und Interaktion zwischen verschiedenen Softwareapplikationen bei der Bearbeitung konkreter Planungsaufgaben [Beetz u. a., 2005, 2006] oder auf die Integration verschiedener heterogener Domänenmodelle [Rüppel u. a., 2006].

*Forschung im
Bauwesen*

7.3.2. BESCHREIBUNG DER ERSCHLIESSUNGSSTRUKTUR MIT OWL

OWL-Dokumente verwenden die Notation der Extended Markup Language (XML), sind somit für Menschen und Maschinen lesbar sowie unter verschie-

Merkmale der OWL

² Zu Spezifikationen und Referenzen zu OWL siehe <http://www.w3.org/2001/sw/>

denen Beteiligten auf der Basis des XML-Standards austauschbar. Ebenfalls auf der Basis des XML-Standards können aus einem OWL-Dokument heraus weitere XML- bzw. OWL-Dokumente, die lokal oder im Netz verfügbar sind, referenziert werden. Die in den referenzierten Dokumenten enthaltenen Definitionen werden dabei importiert und können im aktuellen Dokument erweitert werden. Dies erlaubt eine flexible und verteilte Erweiterung des ES-Schemas und Beschreibung von ES-Daten.

OWL-Werkzeuge Zur Erstellung und Bearbeitung von OWL-Dokumenten sind Editoren als freie Software verfügbar³. Darüber hinaus existieren Softwarebibliotheken, die das Einlesen und Verarbeiten von OWL-Dokumenten in Java-Anwendungen ermöglichen⁴. Vor diesem Hintergrund wurde OWL verwendet, um exemplarisch das Datenschema der Erschließungsstruktur formal zu beschreiben, darauf aufsetzend eine Ausprägung der Erschließungsstruktur für ein Gebäude zu erstellen sowie ein exemplarisches Klassifikationssystem für Raum- und Bauteiltypen zu definieren. Diese drei Bestandteile wurden in jeweils separaten OWL-Dokumenten beschrieben, die wie in Abbildung 7.15 dargestellt, organisiert sind.

Umsetzung In der Ausprägung einer Erschließungsstruktur (ES-Daten) werden jeweils die für das konkrete Bauwerk definierten Informationscontainer, Beziehungen zwischen diesen sowie Referenzen auf die einem Container zugeordneten Informationen verwaltet. Diese Informationen selbst, werden jedoch – im Sinne des OHS-Ansatzes – nicht als Bestandteil der Erschließungsstruktur, sondern in Form separater und durch die Erschließungsstruktur referenzierter Inhalte verwaltet. Im Rahmen einer Java-basierten Umsetzung kann die Verwaltung beschreibender Informationen folgendermaßen realisiert werden (Anteile wurden jeweils exemplarisch umgesetzt, siehe Anhang D):

³ wie z. B. der Protégé Ontology Editor, siehe <http://protege.stanford.edu/>

⁴ wie das Jena Semantic Web Framework, siehe <http://jena.sourceforge.net/>

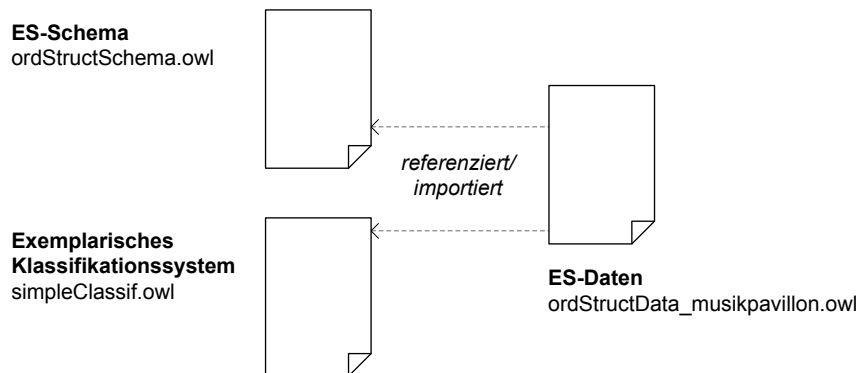


Abbildung 7.15.: Organisation der OWL-Dokumente zur Repräsentation der Erschließungsstruktur

- **Klassifikationssystem**

Für Gebäude als Unterklasse von Bauwerken wurden exemplarisch Bauteiltypen und Typen räumlicher Elemente in einer eigenständigen Ontologie (`simpleClassif.owl`) definiert. Denkbar ist jedoch auch die Bezugnahme auf bestehende Klassifikationssysteme im Bauwesen wie beispielsweise bauwesenspezifische Implementierungen des IFD⁵.

- **3D-Szeneobjekte**

Zur Erstellung, Verwaltung und Präsentation von interaktiven 3D-Szenen fand in der exemplarischen Umsetzung das Java3D-Framework⁶ Anwendung. Mit dessen Hilfe können 3D-Szenen als objektorientierte Modelle in Java-Anwendungen integriert werden. Eine entsprechende Unterklasse (`J3DVisualObjectRef`) der Knotenklasse `VisualObjectRef` wurde so definiert, dass Exemplare dieser Klasse jeweils auf eine Szenegraph-Datei sowie ein Geometrieobjekt innerhalb dieser Datei verweisen. Parallel dazu erlaubt ein entsprechender Dienst die Aus-

⁵ wie [BARBi Project] bzw. die IFD Library, siehe http://www.ifd-library.org/index.php/Home_Page

⁶ siehe <http://java.sun.com/javase/technologies/desktop/java3d/>

wertung von `J3DVisualObjectRef`-Exemplaren und die Darstellung der zugehörigen 3D-Szene.

- **Domänenmodelldaten**

Zur Verwaltung einzelner Domänenmodelle fand ein Dynamik unterstützendes Modellverwaltungssystem⁷ (MVS) Anwendung. Die Referenzierung von Domänenmodelldaten erfolgt über Exemplare der Klasse `BMDateRef`, die Auswertung wiederum über einen entsprechenden Dienst, der in der Lage ist, über die Schnittstelle des MVS auf die verwalteten Daten zuzugreifen.

7.4. DISKUSSION

Zwischenergebnis In diesem Kapitel wurde ein konzeptuelles Schema zur Abbildung von Erschließungsstrukturen für digitale Bauwerksdokumentationen entwickelt. Dabei stellen Informationscontainer und Eigenschaftslinks die zentralen Konzepte solcher Erschließungsstrukturen dar. Mit ihrer Hilfe können einerseits die verschiedenen fachspezifischen Repräsentationen eines Realweltobjektes unter einem zentralen Einstiegspunkt zugänglich gemacht werden, andererseits kann das Extrahieren von Informationen aus komplexen fachspezifischen Repräsentationen mit Hilfe vordefinierter Anfragen erleichtert werden.

Flexible Referenzierung Das vorgeschlagene Konzept der Informationscontainer erlaubt die flexible Referenzierung und damit die logische Zusammenfassung verschiedenartig repräsentierter und verwalteter Informationen unter einem jeweils ein Realweltobjekt vertretenden Identifikator. Innerhalb der beschriebenen Datenstruktur wurden verschiedene Arten von Referenzknoten zum Verweis auf extern verwaltete Informationen vorgeschlagen. Unter Verwendung

⁷ nach dem in [Hauschild, 2003] beschriebenen Ansatz

dieser Referenzknotentypen kann jedem Identifikator eine beliebige Anzahl solcher Verweise zugeordnet werden, wobei Identität, Bedeutung und strukturelle Beziehungen der Informationscontainer frei festgelegt und auch im Nachhinein beliebig modifiziert werden können.

Die Struktur der Informationscontainer trifft keine Annahmen bezüglich weiterer, später hinzuzufügender Typen von Referenzknoten. Auf diese Weise ist eine Erweiterung des ES-Schemas um neue Knotentypen jederzeit möglich, ohne dass dadurch bestehende Ausprägungen des Schemas (bestehende Informationscontainer) beeinflusst werden. Die neu hinzugefügten Knotentypen können dabei auch von bereits bestehenden Containern zur Referenzierung externer Informationen verwendet werden.

Erweiterbare Struktur

Da die Identität und Bedeutung der Realweltobjekte nicht *innerhalb* der Informationscontainer, sondern durch Referenzierung entsprechender Informationen beschrieben werden, ist die vorgeschlagene Lösung nicht auf *Gebäude* als den in der Arbeit zunächst primär betrachteten Bauwerkstyp beschränkt. Für den jeweils zu beschreibenden Bauwerkstyp ist die externe Definition der entsprechenden Elementtypen innerhalb eines Vokabulars und die Beschreibung der Szeneobjekte sowie deren Verknüpfung mit den Informationscontainern erforderlich.

*Übertragbarkeit auf
andere
Bauwerkstypen*

Prozesse der Ausprägung des ES-Schemas, d. h. des Anlegens einer entsprechenden Erschließungsstruktur für eine konkrete Bauwerksdokumentation, wurden zunächst nicht näher betrachtet. Diesbezüglich sind verschiedene Ansätze denkbar.

*Bereitstellung von
ES-Daten*

Aufgrund der Flexibilität der Informationscontainer-Struktur eignet sich diese insbesondere für die Verwendung zur entwurfsbegleitenden Strukturierung externalisierter Repräsentationen des Entwurfsgegenstandes, wie in [Steinmann, 1997] gefordert. Somit kann parallel zum Entwurf für ein Neubauprojekt bereits die objektbezogene Ordnungsstruktur aufgebaut und

*Entwurfsbegleitender
Aufbau*

während der folgenden Planungsphasen schrittweise mit weiteren fachspezifischen Repräsentationen verknüpft werden. Zu diesem Zweck müssen den Entwerfenden/Planenden jeweils entsprechende Editoren zur Verfügung gestellt werden, deren Konzeption jedoch nicht Gegenstand dieser Arbeit sein soll.

*Überführung aus
anderen
Domänenmodellen* Ein großer Teil der zur Erzeugung von ES-Daten, einschließlich der Identität, Bedeutung und Struktur beschreibenden Informationen ist meist ohnehin in den fachspezifischen Repräsentationen enthalten. So bilden architekturenspezifische Modelle das baulich-räumliche Gefüge einschließlich geometrischer Repräsentationen der einzelnen Elemente ab. Diese Informationen können theoretisch auf der Basis des in [Willenbacher, 2002] vorgeschlagenen verknüpfungsbasierten Ansatzes zu einer automatisierten Erzeugung von ES-Daten, respektive externer, beschreibender Informationen verwendet werden. Diese Möglichkeit wurde jedoch nicht weiter untersucht.

8. KOORDINATION VON RECHERCHEMODULEN

Im vorangegangenen Kapitel wurde ein Datenmodell in Form eines konzeptuellen, objektorientierten Schemas beschrieben, das in der Lage ist, Erschließungsstrukturen für beliebige Bauwerke abzubilden. Diese Erschließung des Informationsraums bildet den ersten Schritt zur Unterstützung entwurfsbezogener Informationssuche in der Bauwerksdokumentation. Der zweite Schritt besteht in der Bereitstellung einer Nutzerschnittstelle, die Informationssuchenden den erschlossenen Informationsraum in interpretierbarer Form präsentiert und ihnen die Interaktion mit dem Informationsraum ermöglicht. In Kapitel 6 wurde der modulare Aufbau einer solchen Nutzerschnittstelle gefordert, der eine spätere Erweiterung um zusätzliche Präsentations- und Interaktionstechniken vorsieht.

Ausgangspunkt

Gegenstand dieses Kapitels ist die Entwicklung eines objektorientierten Softwaresystems, das die geforderte Nutzerschnittstelle realisiert. Das System wird mit Hilfe einer formalen Notation implementierungsunabhängig beschrieben. Kritische Teile wurden in Form exemplarischer Prototypen realisiert. Anhand dieser werden zum Abschluss des Kapitels Möglichkeiten zur Implementierung des beschriebenen Systems vorgestellt.

Zielsetzung

Aufgrund der vorgesehenen Erweiterbarkeit ist eine vollständige Beschreibung des Systems nicht möglich. Stattdessen stehen in erster Linie Eigenschaften wie Struktur/Aufbau und Basisfunktionalitäten des Systems im Vordergrund, die für die Koordination und Integration später einzufü-

Abgrenzung

gender Komponenten erforderlich sind. Dabei wird ein Grundgerüst bzw. *Framework* für die gesuchte Nutzerschnittstelle entwickelt. Dieses wird im Folgenden als **Rechercheframework** bezeichnet. Die Auswahl bzw. Konzeption und Realisierung von Präsentations- und Interaktionsfunktionalität ist dabei Aufgabe der einzelnen **Recherchemodule** und liegt nicht mehr im Fokus dieser Arbeit. Mögliche Lösungsansätze werden jedoch zum Abschluss des Kapitels diskutiert.

Inhalt Dieses Kapitel umfasst folgende Schwerpunkte:

- Illustration der Benutzung der „vollständigen“ Informationsumgebung aus der Sicht eines Nutzers in Form eines exemplarischen Szenarios
- Überblick zu grundlegenden Vorgehensweisen bzw. Ansätzen der objektorientierten Softwareentwicklung, insbesondere Ansätzen zur Wiederverwendung von Entwurfswissen
- Entwicklung und formale Spezifikation des Rechercheframeworks
- Vorschlag zur Implementierung.

Der folgende Abschnitt erläutert anhand eines typischen Entwurfsszenarios die Benutzung einer digitalen Bauwerksdokumentation, wobei die Sicht eines Informationssuchenden beim Umgang mit der gesuchten Nutzerschnittstelle beschrieben wird. Gegenstand des Szenarios ist das in Abbildung 8.1 dargestellte Bürogebäude.

8.1. EXEMPLARISCHES SZENARIO EINER RECHERCHESITZUNG

Ausgangssituation Für ein Entwurfsszenario, in dem die Rechercheumgebung zum Einsatz kommt, wird folgende Situation angenommen:

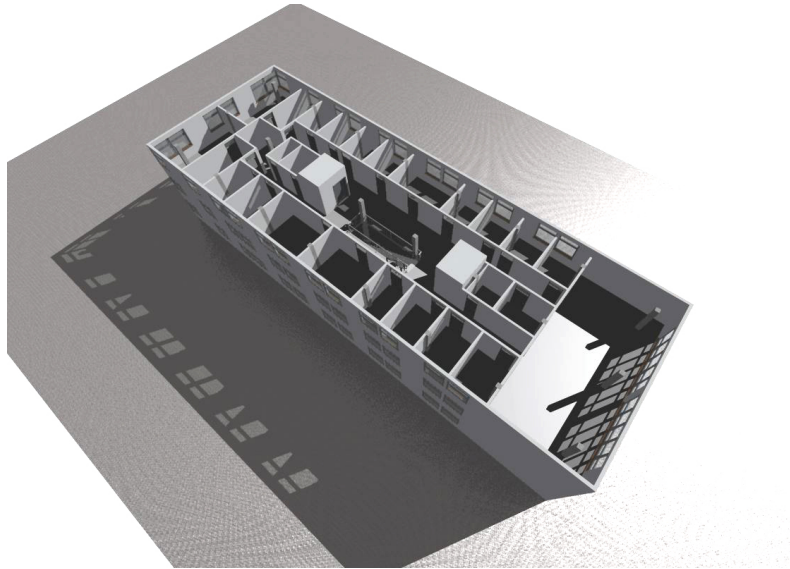


Abbildung 8.1.: Exemplarisch betrachtetes Bürogebäude

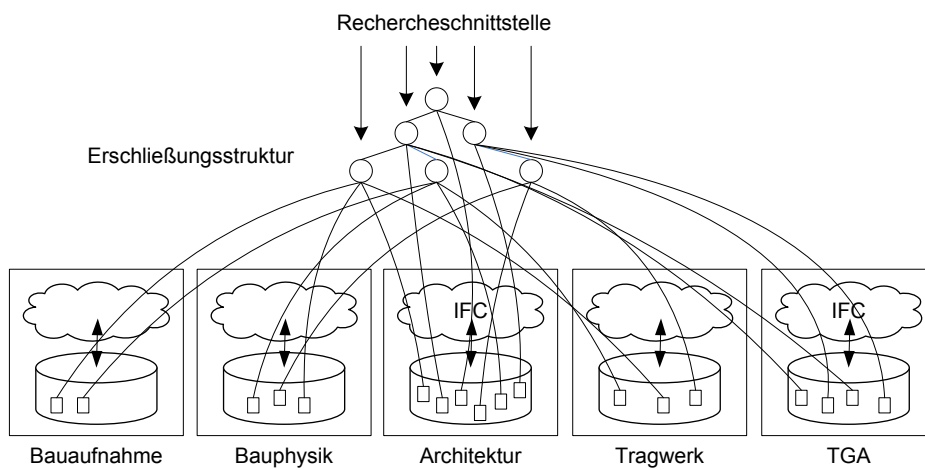


Abbildung 8.2.: Verfügbarer Informationsraum des Szenarios

Entwurfsaufgabe In einem bestehenden Bürogebäude soll ein Bereich für kleinere Konferenzen geschaffen werden. Dafür werden mehrere größere Tagungsräume benötigt. Darüber hinaus soll mit einer Cafeteria ein Aufenthalts- und Pausenbereich geschaffen werden, über den Pausenverpflegung in kleinerem Umfang möglich ist. Die bestehende Konstellation des Modellverbundes, der aus der Neubauphase des Bürogebäudes übernommen wurde, enthält folgende Domänenmodelle:

- Datenbestand*
- *Architektur* (verwendet das IFC-Schema)
 - *Tragwerk* (Schema unbekannt)
 - *Bauphysik* (Schema unbekannt)
 - *TGA* (verwendet das IFC-Schema).

Aufgrund einer im Lauf der Nutzung durchgeführten baulichen Veränderung besteht für einen Teil des Gebäudes außerdem ein Modell *Bauaufnahme*, dessen Schema unbekannt ist.

Erschlossener Informationsraum Entsprechend des ES-Schemas wurde eine Erschließungsstruktur für die Bauwerksdokumentation des Bürogebäudes angelegt, mit der die Inhalte der fünf Domänenmodelle verknüpft wurden.

Nutzungskontext Ein Architekt (A) ist mit der Entwicklung eines Gesamtkonzeptes für das Projekt beschäftigt und untersucht in diesem Zusammenhang das 2. Obergeschoss des Gebäudes. Im Rahmen der entwurfsbezogenen Informationssuche sollte für den Architekten die Interaktion mit der Bauwerksdokumentation entsprechend der folgenden Schritte möglich sein.

Start der Sitzung *Schritt 1* A startet die Informationsumgebung für eine neue Sitzung zum betreffenden Projekt. Das System zeigt eine Übersicht, die über den Status des verfügbaren Datenbestandes, d. h. die verfügbaren Domänenmodelle und Eigenschaftslinks, sowie über verfügbare Arten von Recherchemodulen informiert (siehe Abbildung 8.3).

Übersicht zur Recheresitzung													
Verfügbare Module	Verfügbare Inhalte für: Bürogebäude Kastanienallee												
<i>Viewer / Browser</i> - 3D-Szene - Aggregationsstruktur <i>Sonstige</i> - Abfragegenerator	<table border="0"> <tr> <td><i>Domänenmodelle</i></td> <td><i>Eigenschaftslinks</i></td> </tr> <tr> <td>Architektur [IFC]</td> <td>Material Dimensionen Bauteiltyp tragende_Bauteile Grundfläche Rauminhalt</td> </tr> <tr> <td>Tragwerk [unbekannt]</td> <td>Lastannahmen gewählte_Dimensionen</td> </tr> <tr> <td>Bauphysik [unbekannt]</td> <td>Wärmebedarf</td> </tr> <tr> <td>Bauaufnahme [unbekannt]</td> <td>dokumentierte_Schäden</td> </tr> <tr> <td>TGA [IFC]</td> <td></td> </tr> </table>	<i>Domänenmodelle</i>	<i>Eigenschaftslinks</i>	Architektur [IFC]	Material Dimensionen Bauteiltyp tragende_Bauteile Grundfläche Rauminhalt	Tragwerk [unbekannt]	Lastannahmen gewählte_Dimensionen	Bauphysik [unbekannt]	Wärmebedarf	Bauaufnahme [unbekannt]	dokumentierte_Schäden	TGA [IFC]	
<i>Domänenmodelle</i>	<i>Eigenschaftslinks</i>												
Architektur [IFC]	Material Dimensionen Bauteiltyp tragende_Bauteile Grundfläche Rauminhalt												
Tragwerk [unbekannt]	Lastannahmen gewählte_Dimensionen												
Bauphysik [unbekannt]	Wärmebedarf												
Bauaufnahme [unbekannt]	dokumentierte_Schäden												
TGA [IFC]													

Abbildung 8.3.: Übersicht der Dokumentation für das *Bürogebäude Kastanienallee*

Start des 1. Viewers *Schritt 2* A startet das Viewersmodul „Aggregationsstruktur“. Dieses stellt den hierarchischen räumlichen Aufbau des Bürogebäudes in Form einer Baumstruktur dar.

Start des 2. Viewers *Schritt 3* A wählt in dieser Baumstruktur das 2. Obergeschoss und startet für dieses das 3D-Viewersmodul. Dieses erzeugt die in Abbildung 8.4 gezeigte Darstellung des 2. Obergeschosses als interaktive 3D-Szene.

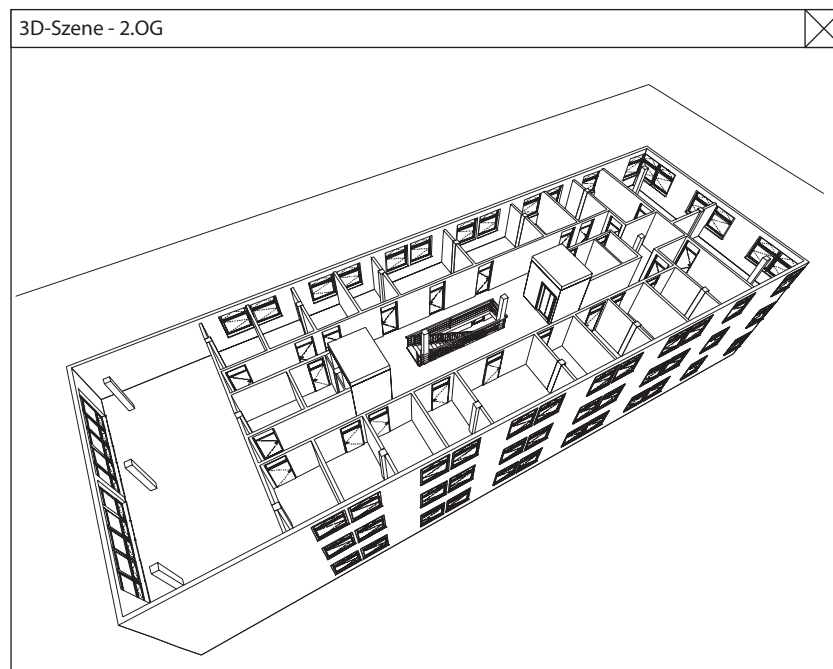


Abbildung 8.4.: 3D-Szene nach dem Start des Viewersmoduls

Untersuchen des Tragsystems I *Schritt 4* A ist mit dem Gebäude noch nicht vertraut und interessiert sich vor allem für die tragenden Bauteile, um so Möglichkeiten für die Neuorganisation der Flächen abschätzen zu können. In der Übersicht wählt A „tragende Bauteile“ und „Elemente in der Grafik kennzeichnen“ aus dem Kontextmenü. Daraufhin erscheint eine Meldung: „Keine korrespondierenden Paare gefunden.“

Grund dafür ist die spezifische Datenstruktur dieser Bauwerksdokumentation: Das Architekturmodell verwendet zwar das IFC-Schema, prägt jedoch nicht alle Klassen aus¹. So existieren in diesem Fall keine Exemplare von `IfcStructuralMember` (vgl. Abschnitt 7.1.2) und auch die entsprechenden Beziehungen `IfcRelAssignsToProduct` wurden nicht ausgeprägt. Stattdessen wurde zur Verwaltung des Tragwerksmodells ein eigenes Domänenmodellschema verwendet und ausgeprägt. A ist mit diesen Tatsachen jedoch nicht vertraut.

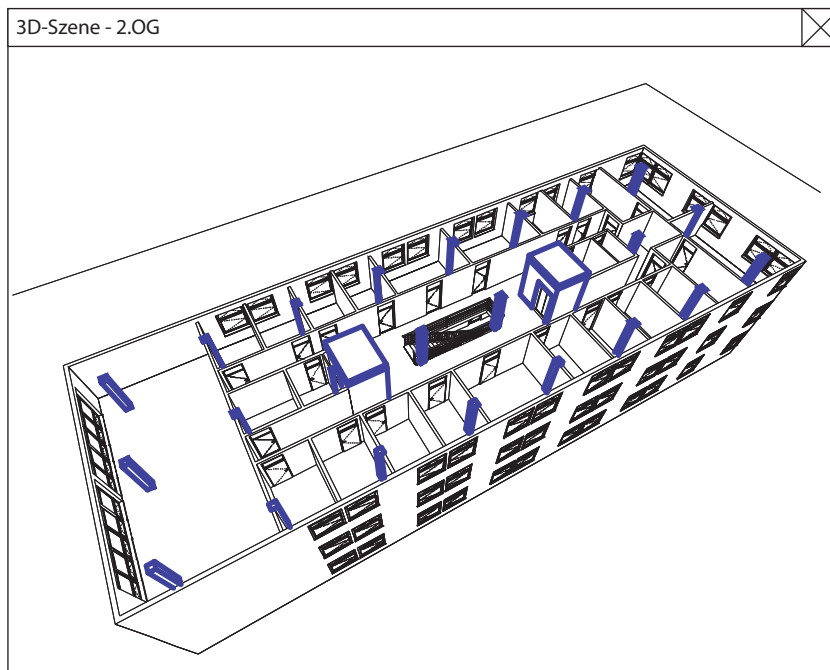


Abbildung 8.5.: 3D-Szene mit Kennzeichnung des Tragsystems

Schritt 5 In der Übersicht existiert jedoch unter „verfügbare Domänenmodelle“ ein Eintrag „Tragwerk“. A wählt diesen Eintrag und wiederum

Untersuchen des Tragsystems II

¹ Dies entspricht der derzeitigen Praxis bezüglich der Implementierung des IFC-Schemas durch Fachapplikationen

„Elemente in der Grafik kennzeichnen“. Daraufhin werden Stützen und Aufzugsschächte in der 3D-Szene eingefärbt, wie in Abbildung 8.5 gezeigt.

Ermitteln baulicher Schäden Schritt 6 A interessiert sich außerdem für bauliche Schäden, die während der Revitalisierungsmaßnahmen zu beheben sind, um gegebenenfalls durch geeignete Maßnahmen Neuorganisation und Reparaturen verbinden zu können. Er wählt in der Übersicht „dokumentierte Schäden“ und „Elemente in der Grafik kennzeichnen“. Daraufhin werden einige Räume eingefärbt (Abbildung 8.6). Schäden wurden offensichtlich raumbezogen erfasst, eine explizite Zuordnung zu einzelnen Bauteilen erfolgte dabei nicht.

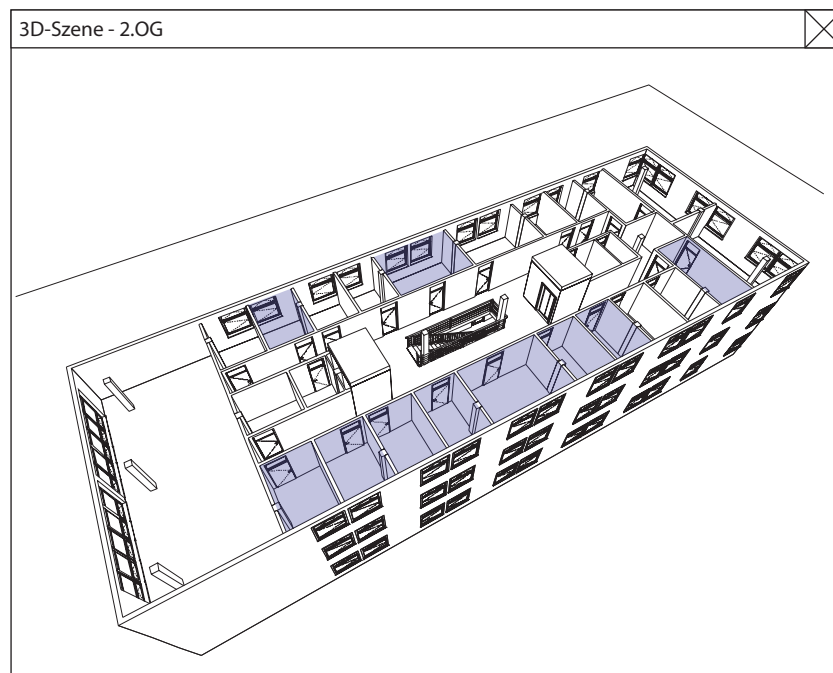


Abbildung 8.6.: 3D-Szene mit Kennzeichnung der Räume mit baulichen Schäden

Schritt 7 A möchte sich im Detail über die für einen speziellen Raum dokumentierten Schäden informieren und selektiert diesen Raum interaktiv in der 3D-Szene. Dann wählt er „Details anzeigen“ aus dem Kontextmenü. Daraufhin wird ein neuer Viewer gestartet, der eine alphanumerische Darstellung der Eigenschaften des Raumobjektes in einem neuen Fenster zeigt (siehe Abbildung 8.7).

Details zu einem Informationscontainer

Objektdetails			
Allgemeine Informationen		Domänenspezifische Repräsentationen	
Objektname:	Raum-208	Architektur	
Objekttyp:	Space	Grundfläche:	9,4 m ²
		Rauminhalt:	22,56 m ³
<i>Beziehungen</i>		Bauphysik	
enthalten_in:	2. Obergeschoss	Wärmebedarf:	>>
enthält:	>>		
wird_begrenzt_von:	>>	Bauaufnahme	
ist_verbunden_mit:	> Raum-207 Flur-200	dokumentierte_Schäden:	>>
<i>Visualisierung verfügbar als</i>			
3D-Szene	>>		
2D-Grundriss	>>		

Abbildung 8.7.: Detailsicht für ein Informationscontainer-Exemplar (schematisch)

Schritt 8 A aktiviert den Pfeil rechts neben „dokumentierte Schäden“. Das System zeigt eine Liste der erfassten, dem Raum zugeordneten Schadensdokumentationen. A wählt eine davon aus, um sie in einem weiteren Detail-Fenster zu öffnen (Abbildung 8.8).

Details einer domänenspezifischen Repräsentation

Schritt 9 A möchte für das gesamte Gebäude die Anzahl der auszutauschenden Fenster ermitteln. Anhand der Informationen im Detail-Fenster zum explorativ gefundenen DamageDoc-Exemplar wäre A nun in der Lage, formale Kriterien für gesuchte Exemplare der Klasse DamageDoc zu spezifizieren:

Formale Anfrage

Datenobjektdetails	
Domänenmodell:	Bauaufnahme
Schemaname:	Survey
Klassenname:	DamageDoc
Objektidentifikator:	obj@8937456009283
<i>Attribute:</i>	
PositionNo.:	208.011
ElementType:	Fenster
Description:	undicht, Rahmen verzogen
<i>Relationen:</i>	
Notes:	Skizze.pdf

Abbildung 8.8.: Detailsicht für ein Datenobjekt eines Domänenmodells (schematisch)

- `ElementType = „Fenster“` UND
- `Description = „*undicht*“`.

Diese könnten für eine Anfrage an das Domänenmodell `Survey` verwendet werden.

8.2. GRUNDLAGEN OBJEKTORIENTIERTER SOFTWARENTWICKLUNG

Die folgenden Abschnitte sollen einen Überblick zu relevanten Grundlagen objektorientierter Softwareentwicklung vermitteln. Auf der einen Seite stehen dabei methodische Vorgehensweisen, auf der anderen Seite Ansätze zur Dokumentation und Wiederverwendung von Entwurfswissen. Für die Entwicklung des gesuchten Systems sind vor allem Letztere von Interesse, da sie die Nachvollziehbarkeit und Begründbarkeit eines Systementwurfs wesentlich stützen können.

8.2.1. METHODEN DER SOFTWARENTWICKLUNG

Als Methode (griechisch: „methodos“) wird im allgemeinen Sprachgebrauch eine systematische Vorgehensweise zur Erreichung eines bestimmten Ziels bezeichnet. In der Softwareentwicklung bezeichnet eine Methode jeweils eine zusammengehörige Einheit aus einer Menge von Konzepten, einer zugehörigen Notation und methodischen Vorgehensweisen [Balzert, 2005].

Methoden

Konzepte sind verallgemeinernde Abstraktionen der in einem Problem- bereich vorkommenden Objekte (vgl. Abschnitt 3.3.1). In der Objektorien- tierung stehen beispielsweise folgende Grundkonzepte im Mittelpunkt:

*Konzepte der
Objektorientierung*

- Objekt - als eine in sich geschlossene Einheit mit Merkmalen und Verhalten
- Klasse - als eine Menge ähnlicher Objekte
- Attribut - als ein Merkmal, über das ein Objekt verfügt
- Methode - als eine Verhaltensweise, über die ein Objekt verfügt
- Botschaft - als Aufforderung an ein Objekt, ein bestimmtes Verhalten „auszuführen“
- Vererbung - als Möglichkeit, neue Klassen zu erzeugen, indem Eigen- schaften und Verhaltensweisen bestehender Klassen übernommen und erweitert werden.

Diese Aufzählung ist lediglich zur exemplarischen Untersetzung des Be- griffs Konzept zu verstehen. Detailliertere Beschreibungen der einzelnen Konzepte der Objektorientierung können beispielsweise [Balzert, 2005] ent- nommen werden.

Methodische Vorgehensweise in der Softwareentwicklung werden durch so genannte Prozess- oder Vorgehensmodelle beschrieben. Sie legen fest, welche Aktivitäten in welcher Reihenfolge auszuführen, welche Ergebnis-

*Methodische
Vorgehensweisen:
Vorgehensmodelle*

se jeweils zu erreichen sind und welche Abhängigkeiten zwischen einzelnen Aktivitäten bzw. Ergebnissen bestehen. Aktivitäten werden jeweils in Phasen zusammengefasst, wobei die verschiedenen existierenden Prozessmodelle unterschiedliche Phaseneinteilungen vornehmen [Balzert, 2001].

Bekannte Vorgehensmodelle Folgende Vorgehensmodelle können als Meilensteine der Beschreibung von Softwareentwicklungsprozessen angesehen werden:

- Wasserfallmodell [Royce, 1970]
- Spiralmodell [Boehm, 1988]
- Unified Process [Kruchten, 1999; Favre, 2003].

Defizite von Vorgehensmodellen Grundsätzlich beschreiben Vorgehensmodelle lediglich einen formalen Ablauf des Entwicklungsprozesses in Form einzelner generischer Arbeitsschritte. Der konkrete Inhalt der Arbeitsschritte, d.h. Entwurfsentscheidungen, deren Hintergründe, Ergebnisse und Konsequenzen sind nicht Gegenstand von Vorgehensmodellen [Schmidt u. Buschmann, 2003]. Somit können sie einen erfahrenen Entwerfer zwar bei der systematischen Dokumentation seines Entwurfs unterstützen, bieten jedoch keine Unterstützung bei der Suche nach für ein Problem angemessenen Entwurfsentscheidungen.

8.2.2. WIEDERVERWENDUNG VON ENTWURFSWISSEN

Parallel zu den Vorgehensmodellen existieren Ansätze zur Wiederverwendung von Entwurfswissen der objektorientierten Softwareentwicklung:

- **Frameworks** ermöglichen die Wiederverwendung von Programmcode zur Implementierung einer Reihe ähnlicher Anwendungen. Sie enthalten das Entwurfswissen implizit in Form von Quellcode.
- **Muster** dokumentieren die wesentlichen Elemente einer Entwurfslösung auf einer implementierungsunabhängigen Ebene, indem sie die

wesentlichen Komponenten, ihre Aufgaben, strukturellen Beziehungen und Interaktionen beschreiben.

FRAMEWORKS

Frameworks sind als ein aus der Praxis der objektorientierten Softwareentwicklung stammender Ansatz zu sehen. Sie wurden in Form vorhandener Artefakte erst im Nachhinein Gegenstand wissenschaftlicher Untersuchungen und Publikationen. Dabei stand vor allem ihre Dokumentation für Anwendungsentwickler im Vordergrund. Siehe dazu [Johnson, 1992, 1997; Demeyer u. a., 2000; Brugali, 2000; Schmidt u. Buschmann, 2003].

*Praxisorientierter
Ansatz*

Johnson verwendet zwei einander ergänzende Definitionen für den Begriff des Frameworks [Johnson, 1997]:

1. Ein Framework ist ein wiederverwendbarer Entwurf eines Systems oder Systemteils, der durch eine Menge abstrakter Klassen und deren Interaktion repräsentiert ist.
2. Ein Framework ist das Skelett einer Anwendung, das durch den Anwendungsentwickler angepasst werden kann.

Aus praktischer Sicht sind Frameworks wiederverwendbare Implementierungen von Funktionalitäten, die alle Anwendungen eines bestimmten Anwendungsbereichs oder einer bestimmten Produktfamilie teilen. Sie stellen eine wiederverwendbare Umgebung für die anwendungsspezifischen Komponenten einer konkreten Applikation bereit und haben sich vor diesem Hintergrund als leistungsfähige Technologie für die Entwicklung

*Wiederverwendung
von Quellcode*

- erweiterbarer,
- flexibler,
- skalierbarer

Software² etabliert [Schmidt u. Buschmann, 2003].

Abstrakte Klassen

Das Schlüsselement von Frameworks ist die Idee der abstrakten Klasse [Johnson, 1997]: Abstrakte Klassen prägen keine Exemplare aus. Das heißt, sie sind nicht als Schablone für die Ausprägung von Objekten zu sehen, sondern als Schablone für spezialisierende Unterklassen. Durch die Definition einer oder mehrerer abstrakter Methoden³ können abstrakte Klassen einheitliche Schnittstellen für alle ihre Unterklassen festlegen. Konkrete Unterklassen müssen diese abstrakten Methoden jeweils implementieren. Optional können abstrakte Klassen Standardimplementierungen für einige ihrer Methoden bereitstellen. Unterklassen können diese dann entweder übernehmen oder entsprechend ihrer Belange überschreiben.

Interaktion mehrerer abstrakter Klassen

Frameworks enthalten in der Regel mehrere abstrakte Klassen, die in einer bestimmten Art und Weise untereinander bzw. mit der vom Framework bereitgestellten Umgebung interagieren. Innerhalb des Frameworks ist die Interaktion der Klassen bereits implementiert oder durch abstrakte Methoden vorgegeben. Anwendungsentwickler können nun die abstrakten Klassen entsprechend ihrer Anforderungen durch konkrete Klassen spezialisieren. Auf diese Weise wird sichergestellt, dass Exemplare der konkreten Anwenderklassen entsprechend der vorgegebenen Interaktionsprinzipien untereinander sowie mit der vom Framework bereitgestellten Umgebung

² Bekannte Vertreter sind z.B. Frameworks für grafische Benutzeroberflächen wie Java Swing (<http://java.sun.com/javase/technologies/desktop/>) oder die Microsoft Foundation Classes ([http://msdn.microsoft.com/en-us/library/d06h2x6e\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/d06h2x6e(VS.80).aspx)) aber auch Frameworks für spezifische Anwendungsbereiche wie das Java Agent Development Framework (<http://jade.tilab.com/>)

³ Abstrakte Methoden besitzen keinen implementierenden Rumpf.

kommunizieren können. Die Klassen eines Frameworks sind somit immer in eine Umgebung eingebunden. Sie können nicht wie die Klassen einfacher Bibliotheken isoliert benutzt werden.

Mit Hilfe von Frameworks ist die Wiederverwendbarkeit von Entwurfswissen auf Implementierungsebene möglich. Muster zielen dagegen auf die Dokumentation von Entwurfswissen auf konzeptueller Ebene, unabhängig von einer konkreten Implementierung.

OBJEKTORIENTIERTE MUSTER

Muster wurden im Zusammenhang mit dem Architekturentwurf in Abschnitt 4.4.3 bereits erwähnt. Sie zielen auf die Dokumentation von Entwurfswissen zur Lösung eines in einem bestimmten Kontext häufig auftretenden Problems. In diesem Sinne stellen sie externalisiertes Expertenwissen dar, dass zwischen verschiedenen Individuen kommuniziert werden kann. Muster erlauben so die Wiederverwendung erfolgreicher Lösungsprinzipien unabhängig von einer konkreten technischen Realisierung, so dass Entwerfende eine allgemeine bzw. funktionale Lösung nicht von Grund auf neu entwickeln, sondern diese ggf. modifizieren und eine entsprechende spezifische Lösung für die konkrete Problemsituation finden müssen. Muster spiegeln so das Denken in Problem-Lösungs-Paaren wider.

Dokumentation von Entwurfswissen

Anfang der 90er Jahre wurde der ursprünglich von Alexander [Alexander u. a., 1995] stammende Musteransatz in die Softwareentwicklung und -dokumentation übertragen, insbesondere vor dem Hintergrund einer guten Dokumentation bestehender Frameworks [Johnson, 1992].

Ursprünge

Die Bedeutung von Mustern wird von verschiedenen Autoren folgendermaßen charakterisiert:

„Eine der großartigen Eigenschaften der Beschreibung von Mustern ist, dass sie die Entwurfsentscheidungen aus dem Bereich vager Intui-

tion herausführen und es Autoren ermöglichen, die sich ergebenden Vor- und Nachteile explizit zu benennen.” [Gamma u. a., 2004, S. 441]

„Patterns are a design documentation technique, which has the characteristic of communicating the reason of a design, not just the result.” [Brugali, 2000, Abschnitt 4]

Charakteristika von Mustern

Muster sind das Ergebnis empirischer Untersuchungen. Jedes Muster widmet sich einem spezifischen, in einem Kontext häufig wiederkehrenden Problem und beschreibt die gemeinsamen Merkmale aller bekannten erfolgreichen Lösungen des Problems [Alexander u. a., 1995]. Muster sind also nicht kontextfrei, sondern stets an einen bestimmten Anwendungsbereich bzw. an eine bestimmte Situation innerhalb dieses Anwendungsbereichs gebunden. Muster haben, wie Alexander u. a. bereits formulieren, immer den Charakter von Hypothesen – und dies in zweifacher Hinsicht: Sie bauen stets auf der Annahme, dass das behandelte Problem von anderen Personen ebenso als solches wahrgenommen wird, und sie implizieren die Vermutung, dass die festgestellten Merkmale der bekannten Lösungen auch für alle zukünftigen, nicht bekannten Lösungen gelten.

Grenzen von Mustern und Musterkatalogen

Die Allgemeingültigkeit und Vollständigkeit im formalen Sinn sind für Muster allerdings nicht nachweisbar. Folgende aus der Beschreibung eines Musters abgeleitete Aussagen müssen nicht wahr sein:

- Alle denkbaren Lösungen des Problems müssen die im Muster beschriebenen Merkmale aufweisen.
- Wenn ein Artefakt die im Muster beschriebenen Merkmale aufweist, dann ist es eine Lösung des Problems.

Ebenso kann nicht nachgewiesen werden, dass eine gegebene Menge von Mustern alle Probleme innerhalb eines Anwendungsbereiches vollständig abdeckt. Aus diesem Grund werden im Bereich der Softwareentwicklung oft die Bezeichnungen *Muster* oder *Musterkatalog* der von Alexander u. a.

verwendeten Bezeichnung *Mustersprache* vorgezogen. Siehe dazu [Johnson, 1992; Gamma u. a., 2004; Buschmann u. a., 1998].

Allerdings sichern Muster nicht automatisch die Entwicklung guter Entwürfe, d.h. sie sind kein Ersatz für menschliche Kreativität, Intuition und Verstandesleistung. Die Identifikation des Problems und das Finden passender Muster, das Übertragen der in den Mustern beschriebenen generischen Merkmale in konkrete Lösungen und das Zusammenfügen dieser (Teil-)Lösungen bleibt ausschließlich in der Verantwortung der Entwerfenden. Diesen Aspekt betonen auch die Autoren der Mustersammlungen [Alexander u. a., 1995; Schmidt u. Buschmann, 2003].

*Kein Ersatz für
menschliche
Verstandesleistungen*

Muster eignen sich jedoch sehr gut zur Dokumentation eines Entwurfs bzw. der im Entwicklungsprozess getroffenen Entwurfsentscheidungen und können damit die Plausibilität, d.h. die Begründbarkeit und Nachvollziehbarkeit des Entwurfsergebnisses gewährleisten [Johnson, 1992; Brugali, 2000].

*Dokumentation von
Entwicklungs-
prozessen*

Gegenwärtig ist eine Vielzahl objektorientierter Muster bekannt. Die erste und bekannteste veröffentlichte Sammlung von Mustern findet sich in [Gamma u. a., 2004]. Von verschiedenen Autoren wurden weitere Musterkataloge veröffentlicht, Klassifikationen vorgeschlagen und die Anwendung von Mustern diskutiert [Buschmann u. a., 1998; Starke, 2002; Horn u. Reinke, 2002; Larman, 2005]. Ein Konsens hinsichtlich einer gemeinsamen Musterklassifikation wurde bislang jedoch nicht gefunden.

*Bekannte Muster und
Musterkataloge*

Im folgenden Abschnitt dienen einige ausgewählte Muster der Herleitung und Beschreibung des vorgeschlagenen Systemkonzeptes.

8.3. SYSTEMKONZEPT DES RECHERCHEFRAMEWORKS

Schema der Entwurfsbeschreibung Die folgenden Abschnitte entwickeln anhand objektorientierter Muster das Konzept des vorgeschlagenen Systems, wobei die Struktur des Systems schrittweise aufgebaut bzw. verfeinert wird. Die Dokumentation der einzelnen Entwicklungsschritte folgt dabei jeweils einer festen Gliederung:

- Beschreibung des zu lösenden Problems
- Diskussion potentiell relevanter Muster
- Auswahl eines Musters und Übertragung der entsprechenden generischen Lösung auf das Systemkonzept.

Eine Kurzbeschreibung der diskutierten Muster kann jeweils Anhang A entnommen werden, auf die entsprechenden Quellen wird verwiesen.

8.3.1. VORÜBERLEGUNGEN

Problem Recherchemodule sollen interagieren können, d.h. untereinander Mengen von Informationscontainern austauschen und sich gegenseitig zur Darstellung derselben auffordern. Schritt 3 des Szenarios (vgl. S. 196) erfordert beispielsweise Modulinteraktionen in dieser Form. In diesem Zusammenhang wird im Folgenden zwischen dem Prinzip der Darstellung und der Art und Weise der Darstellung unterschieden.

Prinzipien der Darstellung Zu den Darstellungsprinzipien werden dabei u.a. 3D-Szenen, Listen oder hierarchische Strukturierungen gezählt. Darstellungsprinzipien sind abhängig von den mit einem Container verknüpften Knotenobjekten. In Tabelle 7.1 (S. 173) wurden mögliche Darstellungsprinzipien zusammengestellt.

Die Art und Weise der Darstellung bezieht sich auf die Verwendung der Container-Menge beim Erzeugen der Darstellung. Folgende Arten sind zu berücksichtigen:

*Art und Weise der
Darstellung*

- Darstellung aller in der Menge enthaltenen Container (nicht enthaltene sind auszublenden, falls sie zuvor angezeigt wurden)
- Ausblenden aller in der Menge enthaltenen Container in der Darstellung einer übergeordneten Menge von Informationscontainern
- Darstellung aller in der Menge enthaltenen Container im Kontext einer übergeordneten Menge von Informationscontainern, z.B. durch Darstellung in einer bestimmten Farbe
- Darstellung von Details zu einzelnen Objekten.

Aufgrund der geforderten Erweiterbarkeit der Recherveschnittstelle sind den einzelnen Modulen die von den jeweils anderen Modulen implementierten Darstellungsprinzipien nicht bekannt. Jedoch müssen die Module untereinander über die jeweils gewünschte Art und Weise der Darstellung kommunizieren können. Dies bedeutet, dass alle Recherchemodule eine einheitliche Schnittstelle bereitstellen müssen, die die Übergabe einer Menge von Informationscontainern zusammen mit einer entsprechenden Darstellungsaufforderung ermöglicht.

*Kommunikation
„unbekannter“
Recherchemodule*

Darüber hinaus muss es möglich sein, mehrere Recherchemodule parallel zur Darstellung derselben Menge von Informationscontainern zu verwenden, um beispielsweise gleichzeitig eine 3D-Szene, eine Grundrissdarstellung und eine sortierte Elementliste für einen Bereich des Bauwerks zu betrachten. Zu diesem Zweck muss eine Koordination der parallel verwendeten Recherchemodule erfolgen hinsichtlich

*Koordination
mehrerer Module*

- der Synchronisation des von mehreren Modulen dargestellten Teilbereichs,

- der Synchronisation farbiger Kennzeichnungen, d.h. der Verwendung der gleichen Farbe für ein und dieselbe zu kennzeichnende Menge von Informationscontainern durch alle beteiligten Module innerhalb einer Rechtersitzung, sowie
- optional der Synchronisation von interaktiv vorgenommenen Selektionen, d.h. Selektionen, die in einem Modul vorgenommen wurden, sind für alle beteiligten Module zu setzen.

8.3.2. STRUKTURIERUNG DER RECHERCHESCHNITTSTELLE

Problem Die Rechterschnittstelle als Ganzes stellt eine interaktive Anwendung dar: Sie soll Informationssuchenden die interaktive Manipulation des Informationsraums ermöglichen. Dabei ist die Realisierung der Präsentations- und Interaktionsfunktionanlität Aufgabe der einzelnen Rechtersmodule. Diese sollen in Form von Plug-Ins in das Framework eingefügt werden. Innerhalb des Frameworks ist dabei die Kommunikation und Interaktion der Module für folgende Aufgaben festzulegen:

- Austausch von Mengen von Informationscontainern
- Starten der verschiedenen Arten der Darstellung einer Menge von Informationscontainern – unabhängig vom Darstellungsprinzip
- Koordination und Synchronisation mehrerer Module für eine gemeinsam dargestellte Menge von Informationscontainern.

Erweiterbarkeit Zukünftige Erweiterungen müssen in verschiedener Hinsicht in das Gesamtkonzept einbezogen werden. Einerseits ist die Ergänzung neuer Rechtersmodule für spezifische Interaktions- oder Darstellungsformen vorgesehen. Andererseits ist aber auch die Erweiterung des Erschließungsstruktur(ES)-Schemas denkbar, d.h. es können neue Knotenklassen zur Referenzierung von Informationen hinzugefügt werden. Diese Knotenklassen be-

nötigen jeweils korrespondierende Anwendungslogik zur Auswertung und Darstellung der referenzierten Informationen, wobei diese durch einzelne Recherchemodule implementiert werden muss.

Grundsätzlich bieten sich zwei Muster als Ansatz für die Strukturierung der gesuchten interaktiven Anwendung an: *Relevante Muster*

- das *Model-View-Controller*-Muster (MVC)
(siehe [Buschmann u. a., 1998, S. 124ff.] bzw. auch Anhang A) sowie
- das *Presentation-Abstraction-Control*-Muster (PAC)
(siehe [Buschmann u. a., 1998, S. 145ff.] bzw. auch Anhang A).

Da das PAC-Muster den Begriff des *Agenten* verwendet, soll zunächst kurz auf diesen Begriff eingegangen und seine Verwendung im Rahmen dieser Arbeit erläutert werden. Eine einführende Übersicht und Diskussion des Themas findet sich beispielsweise in [Willenbacher, 2002]. Die folgenden Erläuterungen beziehen sich ausschließlich auf Softwareagenten, d. h. Agenten in Form von ausführbaren Softwarekomponenten. Eine allgemein anerkannte Definition für den Begriff des Softwareagenten existiert derzeit nicht [Willenbacher, 2002]. Die meisten Autoren nennen jedoch übereinstimmend die folgenden Eigenschaften eines Softwareagenten [Ferber, 1995; Caglayan u. Harrison, 1998; Wooldridge u. Jennings, 1995; Bellifemine u. a., 2003]: *Softwareagenten*

- **Autonomie**
Agenten führen Aufgaben selbständig und ohne direkte Intervention des Anwenders aus, wobei sie zu einem gewissen Grad selbst die Kontrolle über den Zeitpunkt und den Umfang der Ausführung besitzen.
- **Pro-Aktivität**
Agenten sind fähig, selbst die Initiative zu ergreifen, anstatt lediglich auf externe Ereignisse zu reagieren.

- **Soziale Fähigkeiten**

Agenten sind in der Lage, mit anderen Agenten zu kommunizieren und zu interagieren. Kommunikationsabläufe dienen dabei der Kooperation bzw. Verhandlung und können aus einem komplexen Nachrichtenaustausch bestehen.

- **Delegation**

Agenten führen im Auftrag des Anwenders oder anderer Agenten Aufgaben aus.

Intelligenz Ferber sowie Caglayan und Harrison betonen darüber hinaus die Eigenschaft der Intelligenz [Ferber, 1995; Caglayan u. Harrison, 1998], wobei diesbezüglich keine klare Definition existiert. Im Zusammenhang mit Intelligenz nennen die o.g. Autoren insbesondere die Fähigkeit eines Agenten, eigenständig Entscheidungen treffen zu können und das eigene Handeln auf die Erfüllung bestimmter Ziele und/oder Neigungen auszurichten. Zur Realisierung dieser Fähigkeiten kommen zumeist Ansätze aus der *künstlichen Intelligenz* (KI) zur Anwendung. Letztlich entscheiden Entwicklungsintention und Anwendungszweck eines Agenten, in welchem Maße KI-Ansätze sinnvoll einbezogen und implementiert werden können. Dies soll jedoch nicht Gegenstand der vorliegenden Arbeit sein.

Agenten im Sinne des PAC-Musters Im Rahmen dieser Arbeit wird der Begriff des Agenten, bzw. Softwareagenten in Orientierung an [Buschmann u. a., 1998] folgendermaßen verwendet: **Agenten** im Sinne des PAC-Musters sind informationsverarbeitende Komponenten, die über Möglichkeiten zum Senden und Empfangen von Nachrichten und Ereignissen (z.B. Benutzereingaben) verfügen. Darüber hinaus verwalten sie ihren Zustand mit Hilfe eigener Datenstrukturen. Ein Prozessor innerhalb des Agenten, im Muster als *Kontrollkomponente* bezeichnet, ist für das Bearbeiten von Nachrichten und Ereignissen, das Aktualisieren des Zustandes und gegebenenfalls das Erzeugen neuer Ereignisse und Versenden neuer Nachrichten verantwortlich.

Für das PAC-Muster sind insbesondere die Autonomie und die sozialen Fähigkeiten eines Agenten von Bedeutung. Technisch wird dies realisiert durch den asynchronen Austausch adressierter Nachrichten über einen zentralen Dienst, wodurch eine lose Kopplung der Agenten erreicht wird. Darüber hinaus obliegt jedem Agenten selbst die individuelle Realisierung seiner Datenverwaltungs- und Präsentationsanteile.

*Autonome, lose
gekoppelte
Komponenten*

Das MVC-Muster weist einige Nachteile auf, die im Konflikt mit den Anforderungen an die Rechercheumgebung stehen: Im MVC-Muster besteht die Annahme eines singulären Modells für alle beteiligten Ansichten. Aus der Sicht der Rechercheumgebung fokussiert die Informationssuche zwar auf *einen* Informationsraum (= Erschließungsstruktur + Modellverbund), Informationssuchende können jedoch während einer Recherche Sitzung unterschiedliche Teilmengen dieses Informationsraums bestimmen und gleichzeitig zur Anzeige bringen. Die Anzahl dieser Teilmengen, die gleichzeitig explorativ untersucht werden können, soll nicht im Voraus auf eine bestimmte Zahl begrenzt werden. Die Annahme eines singulären, von allen Ansichten gleichermaßen verwendeten Modells ist daher unzulässig. Darüber hinaus hätte durch die im MVC-Muster vorgesehene enge Kopplung zwischen Modell und Ansichten bzw. Eingabekomponenten eine Erweiterung des ES-Schemas Auswirkungen auf alle bestehenden Viewer-Komponenten bzw. Recherchemodule, da diesen das ES-Schema (= Modell) bekannt sein muss.

*Diskussion der
Muster*

Aus den genannten Gründen wurde das PAC-Muster als Strukturierungsansatz vorgezogen. Durch eine bedachte Zerlegung der Rechercheumgebung in einzelne Agenten bzw. das Vermeiden einer Zerlegung in zu viele Agenten wirken sich die negativen Aspekte dieses Musters nicht zwangsweise nachteilig aus.

Wahl des Musters

Das PAC-Muster sieht die Strukturierung einer interaktiven Anwendung in Form einer erweiterbaren Hierarchie kooperierender Agenten vor. Als

*Erweiterbare
Hierarchie von
Agenten*

konkreter Lösungsansatz ergibt sich dabei die in Abbildung 8.9 zunächst informal dargestellte Grundstruktur des Rechercheframeworks. Gestrichelt dargestellte Anteile repräsentieren jeweils mögliche Erweiterungen der Struktur zur Laufzeit.

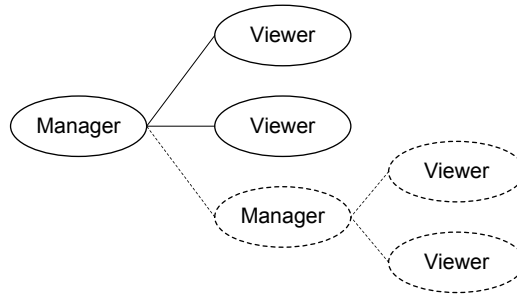


Abbildung 8.9.: Informale Darstellung der Viewer-Manager-Struktur

Manager und Viewer

Jeweils ein Manageragent koordiniert mehrere Vieweragenten, die jeweils gemeinsam einen vom Informationssuchenden gewählten Teilbereich darstellen. Jeder Manageragent kennt die jeweils darzustellende Menge von Informationscontainern sowie die daraus aktuell selektierten Container und benachrichtigt die angemeldeten Viewer über Änderungen bzgl. dieser beiden Werte. Vieweragenten bieten Darstellungsfunktionalität in Form von Diensten an, die andere Agenten in Anspruch nehmen können. Die Vieweragenten spezifizieren jeweils in einer Beschreibung der angebotenen Dienste, welches Darstellungsprinzip sie konkret realisieren und gegebenenfalls welche zusätzlichen Optionen sie für einzelne Dienste anbieten. Manageragenten führen eine Liste aller von Viewern angebotenen Dienste und stellen diese Informationssuchenden *und* Vieweragenten zur Auswahl sowie zur Anforderung der Dienste zur Verfügung. Der Manageragent an der Wurzel der Hierarchie ist darüber hinaus verantwortlich für die Verwaltung von Informationen, die potentiell für alle Agenten innerhalb einer Recheresitzung relevant sind. Dazu zählt beispielsweise eine Liste aller vom Informations-

suchenden definierten Farbzuordnungen zu Mengen von Informationscontainern in Form einer für alle Module gültigen Legende.

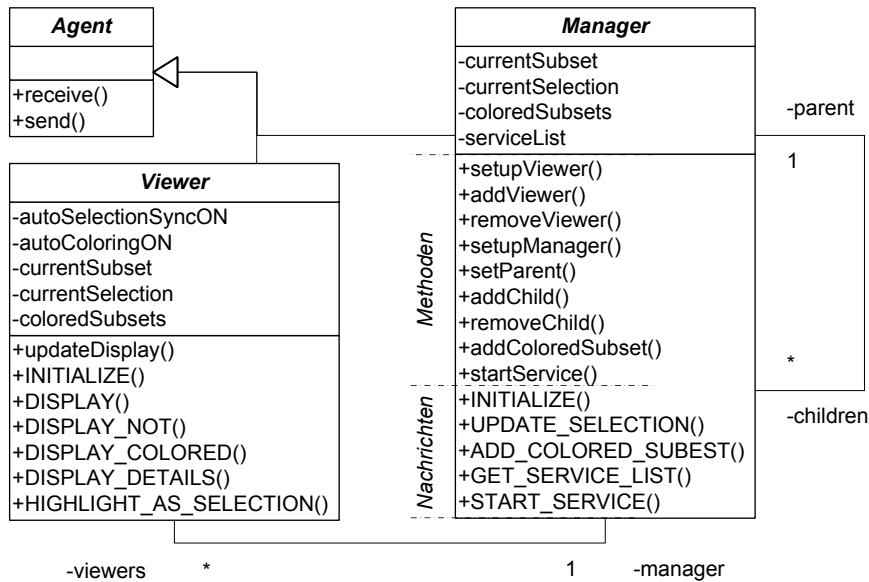


Abbildung 8.10.: Klassendiagramm der Viewer-Manager-Struktur

Das Klassendiagramm in Abbildung 8.10 zeigt die Beziehungsstrukturen zwischen Manager- und Vieweragenten, die jeweils zu verwaltenden Daten, bereitgestellte Methoden und die für den jeweiligen Agententyp gültigen Nachrichten. Großbuchstaben kennzeichnen jeweils die Namen von Nachrichten, Methodennamen verwenden Kleinbuchstaben. Nachrichten werden grundsätzlich über die in der Klasse **Agent** definierten Schnittstellen ausgetauscht und ermöglichen so die lose Kopplung der Komponenten.

Statische Struktur

Die folgenden Sequenzdiagramme zeigen die Interaktion der einzelnen Agenten für einige Schritte des in Abschnitt 8.1 beschriebenen exemplarischen Szenarios.

Interaktionen

In Schritt 1 wurde die Informationsumgebung gestartet; Schritt 2 bestand im Starten des ersten Viewers (vgl. S. 194). Zu Beginn des Schrittes 2 ist

Start des 1. Viewers

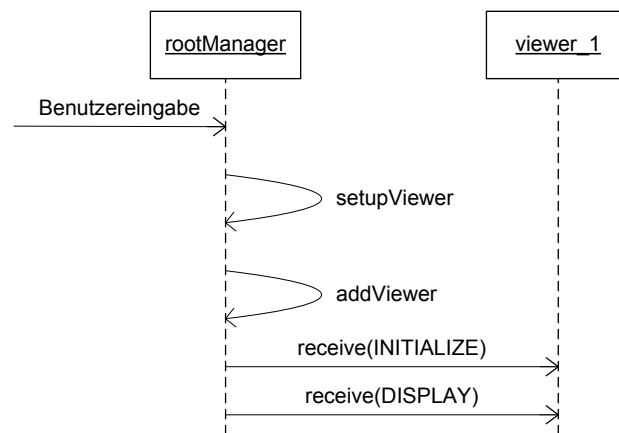


Abbildung 8.11.: Sequenzdiagramm: Starten des ersten Viewer-Agenten

zunächst nur der Manageragent an der Wurzel der Hierarchie aktiv. Dessen Benutzeroberfläche realisiert gleichzeitig die im Szenario genannte Übersicht der Recheresitzung. Über diese können Informationssuchende Teilbereiche wählen und Vieweragenten für diese Teilbereiche starten. Erfolgt keine explizite Selektion eines Teilbereiches, werden die Viewer automatisch für den gesamten Informationsraum, d. h. die Menge aller verfügbaren Informationscontainer gestartet. Das Sequenzdiagramm in Abbildung 8.11 zeigt den Ablauf beim Starten des ersten Viewers.

Start des 2. Viewers

In Schritt 3 (vgl. S. 196) wählt der Informationssuchende das 2. Obergeschoss in der Aggregationsstruktur (= Darstellung des 1. Viewers) und startet aus dem Viewer heraus die Darstellung des Geschosses in einem zweiten Viewer. Das Starten des zweiten Viewers für diese interaktiv selektierte Menge von Informationscontainern einschließlich des Startens eines zusätzlichen Manageragenten für diesen neuen Teilbereich ist im Sequenzdiagramm in Abbildung 8.12 dargestellt.

Untersuchen des Tragsystems/farbige Kennzeichnung

Während der Informationssuchende das Tragsystem des Bürogebäudes untersucht, müssen einzelne Informationscontainer in der Darstellung far-

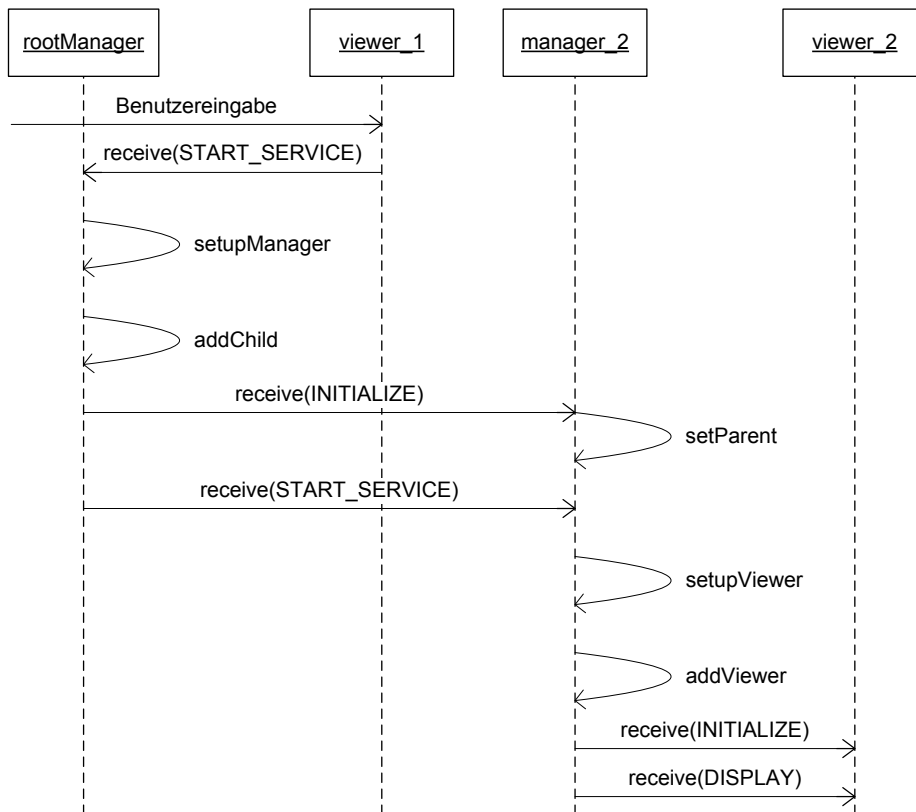


Abbildung 8.12.: Sequenzdiagramm: Starten eines Viewers und eines neuen Managers für eine gewählte Teilmenge

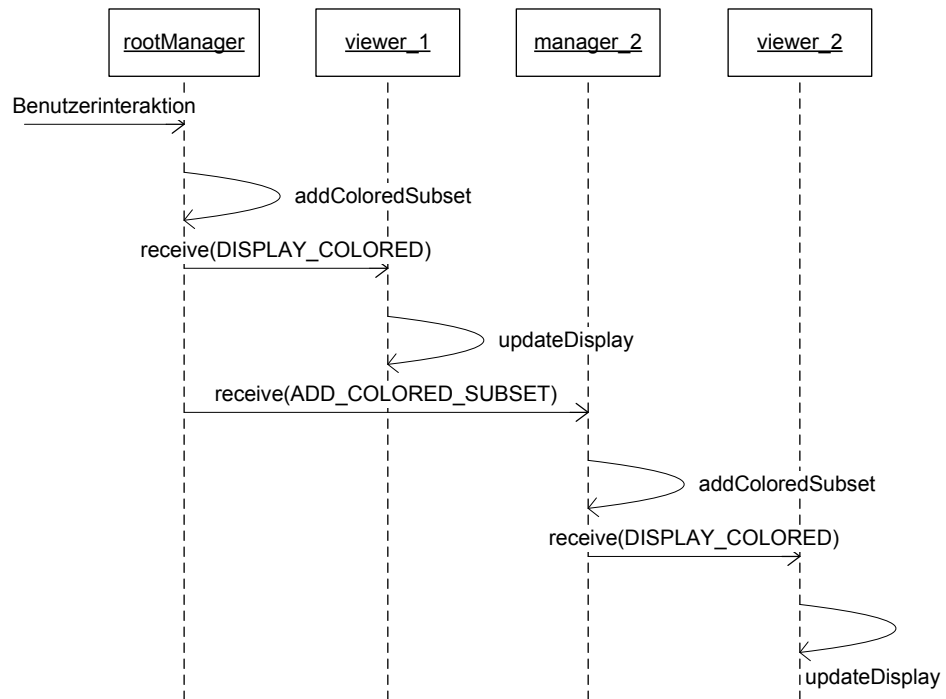


Abbildung 8.13.: Sequenzdiagramm: Erzeugen und Darstellen einer Menge farbig zu kennzeichnender Informationscontainer

big gekennzeichnet werden (Schritt 5, S. 197). Abbildung 8.13 zeigt die Interaktion der Agenten während des Erzeugens einer neuen farblich zu kennzeichnenden Menge von Informationscontainern und die Weitergabe an alle Viewer- und Manageragenten.

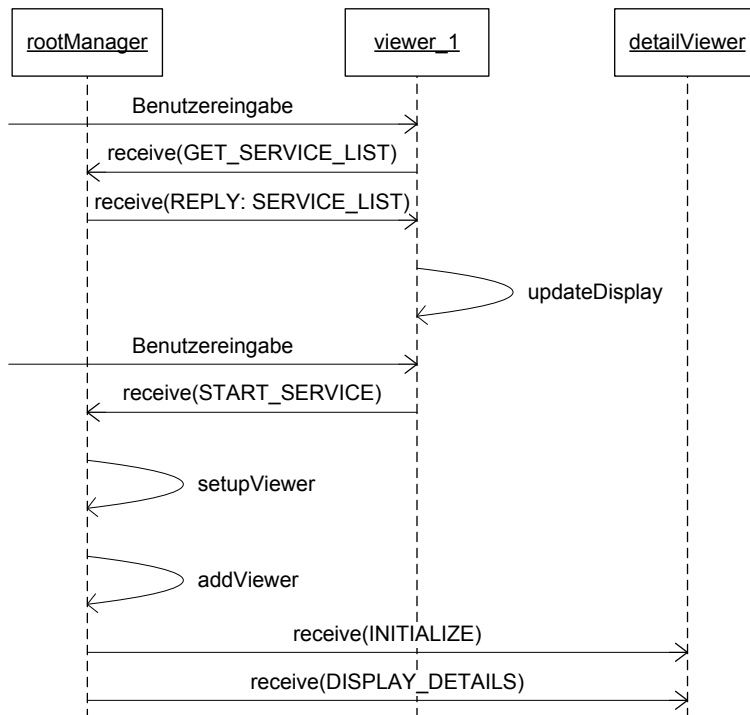


Abbildung 8.14.: Sequenzdiagramm: Starten einer Detail-Darstellung zu einem einzelnen Informationscontainer

In Schritt 7 (vgl. S. 198) des Szenarios wählt der Informationssuchende interaktiv einen einzelnen Informationscontainer (eine Raum)) und ruft eine Detaildarstellung für diesen Container auf. Die zugehörige Agenten-Interaktion ist im Diagramm in Abbildung 8.14 dargestellt.

Details zu einem Informationscontainer

Der aktive Viewer muss zunächst vom Manager die aktuelle Liste der verfügbaren Darstellungsdienste ermitteln, um diese – z.B. über ein Kon-

textmenü – dem Nutzer zur Auswahl anzubieten. Nachdem der Nutzer einen Dienst gewählt hat, wird beim Manager das Aktivieren eines entsprechenden Agenten angefordert.

8.3.3. ERZEUGEN VON VIEWEREXEMPLAREN UND INTEGRATION NEUER VIEWERKLASSEN

Problem Während einer Rechercsesitzung sollen bei Bedarf mehrere Exemplare eines
Viewer als Agenten Viewertyps erzeugt und gleichzeitig verwendet werden können. Dabei spielt es keine Rolle, ob die Exemplare jeweils verschiedene Teilbereiche oder ein und denselben Teilbereich darstellen. Beispielsweise könnte ein Informationssuchender eine 3D-Szene des gesamten Gebäudes und gleichzeitig eine weitere 3D-Szene für einen einzelnen Raum betrachten. Ebenso könnte das gesamte Gebäude gleichzeitig aus mehreren Blickwinkeln betrachtet werden. Die einzelnen Viewerexemplare werden an unterschiedlichen Stellen innerhalb der Agentenstruktur benötigt, d.h. sie kooperieren jeweils mit unterschiedlichen Manageragenten. Dafür muss jedes Viewerexemplar die für einen Agenten im Sinne des PAC-Musters erforderliche Kommunikationsfähigkeit besitzen. Somit ist jedes Viewerexemplar als eigenständiger Agent zu betrachten.

*Integration neuer
Viewerklassen* Neue konkrete Viewerklassen müssen in die Rechercheumgebung eingefügt werden können, ohne dass ein Rekompilieren des gesamten Systems notwendig ist. Nach Möglichkeit soll eine Anmeldung neuer Viewerklassen zur Laufzeit erfolgen. Das Wissen bezüglich verfügbarer Viewertypen ist somit laufzeitspezifisch und das Framework sollte dahingehend keine Annahmen treffen. Stattdessen sollten an zentraler Stelle innerhalb des Frameworks die Registrierung von Viewertypen zur Laufzeit und sinnvollerweise zugleich auch Mechanismen für ihre Ausprägung vorgesehen werden.

Relevante Muster Für diese Problemstellung bieten sich die beiden Muster

- *Fabrikmethode* (siehe [Gamma u. a., 2004, S. 131 ff.] bzw. Anhang A) und
- *Prototyp* (siehe [Gamma u. a., 2004, S. 144 ff.] bzw. Anhang A)

an.

Beide Muster adressieren die Erzeugung von Exemplaren konkreter anwendungsspezifischer Klassen aus einem Framework heraus, wobei abstrakte Frameworkklassen lediglich eine Schnittstelle zur Objekterzeugung definieren, die von konkreten Klassen zu implementieren ist. Der Unterschied der beiden Muster liegt in der Anzahl der jeweils benötigten konkreten Klassen je zu erzeugendem Objekttyp. Das Fabrikmethode-Muster erfordert für jede konkrete Produktklasse eine korrespondierende konkrete erzeugende Klasse. Ein Produktexemplar wird jedoch nur auf Anfrage erzeugt. Für das Prototyp-Muster wird ausschließlich die konkrete Produktklasse benötigt. Diese wird zur Laufzeit in Form eines prototypischen Exemplars verwendet, das jeweils vorhanden sein muss.

*Diskussion der
Muster
Prinzip*

Grundsätzlich muss die gemeinsame Oberklasse **Agent** alle zur Kommunikation und Interaktion der Agenten, d. h. auch zum Einfügen neuer Agenten, benötigten Schnittstellen spezifizieren. Daher könnten sowohl neue Fabrikagenten als auch neue prototypische Vieweragenten problemlos in die zur Laufzeit ausgeprägte Kooperationshierarchie der Viewer- und Manageragenten eingefügt werden.

*Anmeldung zur
Laufzeit*

Im Fall des Fabrikmethode-Musters müssen dann jeweils 2 Agentenklassen implementiert werden: Eine, die den benötigten Viewertyp repräsentiert (das Produkt), und eine, die den „Hersteller“ des Produktes repräsentiert. Von dieser wird jeweils ein Exemplar zur Laufzeit registriert. Alternativ könnte auch lediglich die jeweils konkrete Viewerklasse einschließlich einer Kopier-Methode implementiert werden. Das erste Viewerexemplar müsste dann als Prototyp zur Laufzeit registriert werden und würde auf Anfrage

*Anforderungen an die
Umsetzung*

Kopien von sich selbst erzeugen. Dies hätte jedoch zur Folge, dass jeweils ein vollständiges Exemplar jedes Viewertyps einschließlich aller Darstellungs- und Interaktionsfunktionalitäten zur Laufzeit vorgehalten werden muss, auch wenn es innerhalb der konkreten Rechercsesitzung vielleicht nicht benötigt wird. Zwar können diese Funktionalitäten für das prototypische Exemplar deaktiviert werden, dies stellt jedoch keine optimale Lösung dar. Darüber hinaus ist das vollständige Kopieren eines Viewerexemplars einschließlich seines Zustandes nicht immer sinnvoll, da der neue Viewer eventuell einem anderen Manager zugeordnet werden muss und einen anderen Teilbereich darstellen soll als das prototypische Exemplar.

Wahl des Musters Vor diesem Hintergrund wird eine Trennung in Fabrik- und Vieweragenten vorgezogen. Die vom Framework bereitzustellende Struktur ist zunächst informal in Abbildung 8.15 dargestellt.

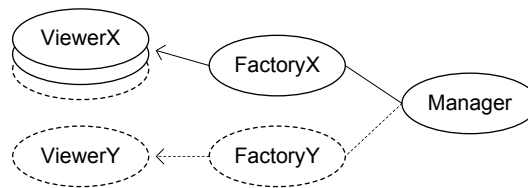


Abbildung 8.15.: Informale Darstellung der Fabrik-Manager-Struktur

Fabrikagenten erzeugen Viewerexemplare Viewerfabriken (Klasse **Factory**) sind neben Viewer- und Manageragenten ein weiterer Agententyp des Frameworks. Sie sind jeweils für die Registrierung eines neuen Viewertyps sowie für die Erzeugung von Exemplaren dieses Typs verantwortlich.

Prozessoren für Knotenklassen Für einige Referenzknotentypen des ES-Schemas ist zur Erzeugung von Detaildarstellungen spezielle Prozessorfunktionalität erforderlich. Zum Aufbau der in Schritt 7 des Szenarios (vgl. S. 198) erwähnten Detailsicht wird z.B. die Bezeichnung des dem Informationscontainer zugeordneten Elementtyps als alphanumerische Zeichenkette benötigt. In Abhängigkeit

vom referenzierten externen Vokabular oder Klassifikationssystem und dessen technischer Verwaltung kann diese Information jedoch in verschiedenen Formaten vorliegen. Ein Prozessor muss somit für die Auswertung von `ClassificationRef`-Exemplaren und die Bereitstellung der benötigten Informationen in einem einheitlichen Format sorgen. Ähnliche Prozessoren werden auch für `BMDataRef`-Exemplare benötigt. Abschnitt 8.3.5 geht detaillierter auf diese Problematik ein.

Vor diesem Hintergrund ist die Trennung von Viewern und Viewerfabriken ebenfalls sinnvoll. Die ohnehin zentral zu registrierenden Viewerfabriken können gleichzeitig die für eine bestimmte Referenzknotenklasse benötigte Prozessorfunktionalität bereitstellen. Diese Funktionalität ist so an zentraler Stelle verfügbar und den Viewern kann ausschließlich das Erzeugen interaktiver Präsentationen übertragen werden.

Fabrikagenten realisieren Prozessorfunktionalität

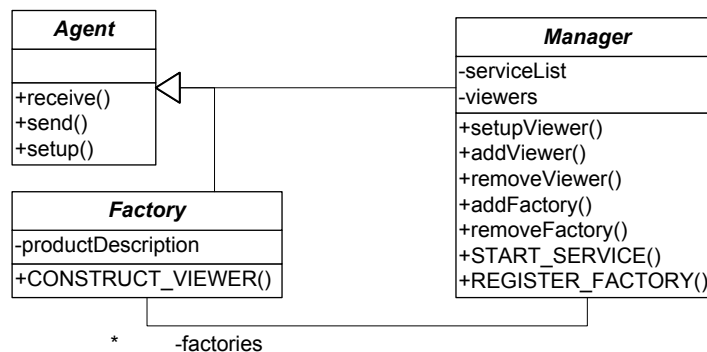


Abbildung 8.16.: Klassendiagramm: Manager- und Fabrikagenten

Das Klassendiagramm in Abbildung 8.16 zeigt die an der Registrierung und Ausprägung neuer Viewertypen beteiligten Klassen. Viewerfabriken (`Factory`) verwalten eine Beschreibung des von ihnen erzeugten Viewertyps einschließlich der Darstellungsdienste, die Viewer dieses Typs anbieten.

Statische Struktur

Über die Nachricht `CONSTRUCT_VIEWER` kann eine Fabrik aufgefordert

Interaktionen

werden, ein neues Viewerexemplar zu erzeugen. Manager führen eine Liste der angemeldeten Viewersfabriken. Neue Viewersfabriken melden sich über REGISTER_FACTORY beim Manager an und übergeben gleichzeitig ihre Produktbeschreibung an den Manager. Erhält ein Manager von die Nachricht START_SERVICE, ermittelt er aus seiner Liste den dem angeforderten Dienst entsprechenden Fabrikagenten und fordert diesen auf, ein neues Viewerexemplar zu erzeugen. Nach der Übergabe des Viewerexemplars fügt er dieses der eigenen Liste aktiver Viewer hinzu, startet dessen Initialisierung und fordert zur Ausführung des angeforderten Darstellungsdienstes auf. Die entsprechenden Interaktionen sind im Sequenzdiagramm in Abbildung 8.17 dargestellt.

Abhängigkeiten Das Registrieren eines neuen Fabrikagenten und das Anfordern eines neuen Viewerexemplars von diesem können als zwei getrennte Sequenzen betrachtet werden, wobei die erste jedoch Voraussetzung für die zweite ist. Zwischen beiden Sequenzen könnten weitere Interaktionen zu inhaltlich unabhängigen Aufgaben stattfinden.

Aufteilung von Managementaufgaben Genau betrachtet ergeben sich innerhalb des Frameworks zwei verschiedene Arten von Managementaufgaben:

- die zentrale Verwaltung organisatorischer Informationen, die für die gesamte Rechtersitzung relevant sind, und
- die dezentrale Koordination von Darstellungsfunktionalität für einzelne darzustellende Teilbereiche.

Management der Rechtersitzung Zu organisatorischen Informationen zählen:

- die Arten der verfügbaren Viewertypen und die von ihnen angebotenen Darstellungsdienste sowie
- die farblich zu kennzeichnenden Mengen von Informationscontainern, die ein Informationssuchender während einer Sitzung definiert hat.

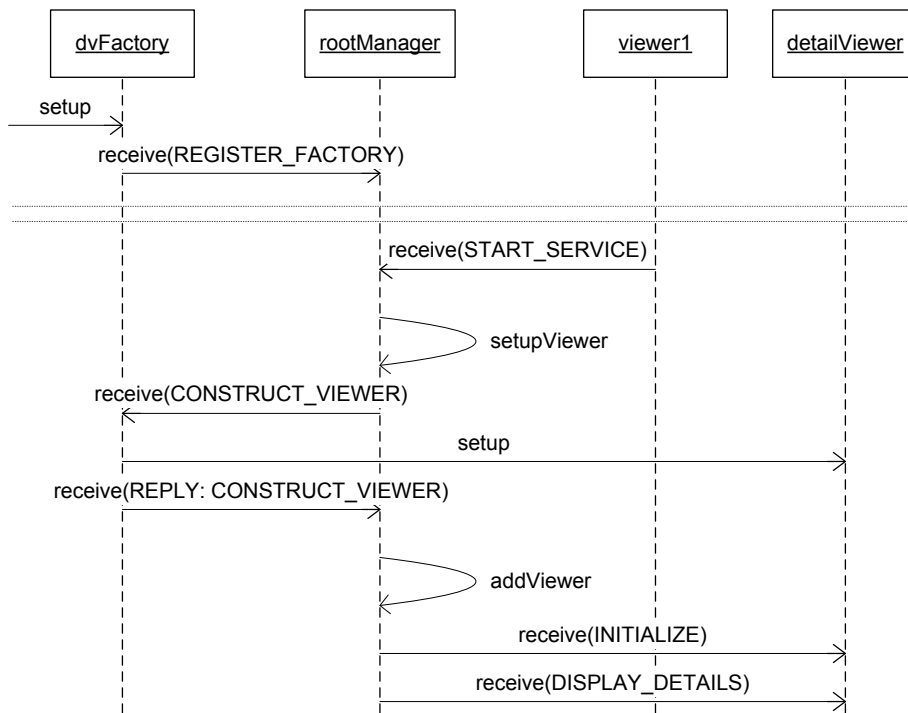


Abbildung 8.17.: Sequenzdiagramm: Anmelden eines neuen Fabrikanten und Erzeugen eines neuen Vieweragenten

Zentrale Informationsverwaltung

Eine zentrale Registratur muss die Anmeldungen neuer Fabrikagenten entgegennehmen und parallel einen Katalog aller bereitstehenden Viewerdienste führen, über den andere Agenten im Auftrag des Informationssuchenden diese Dienste anfordern können. Ebenfalls an zentraler Stelle muss eine Liste der farbigen Kennzeichnungen geführt und bei Änderungen derselben die Benachrichtigung der abonnierenden Agenten ausgelöst werden. Darüber hinaus ist es technisch sinnvoll, eine Liste aller an der Sitzung beteiligten Agenten zu führen, um diese beim Beenden der Sitzung korrekt deaktivieren und beenden zu können. Zur Ausführung dieser Aufgaben wird für jede Rechtersitzung jeweils nur ein Exemplar einer entsprechenden Klasse benötigt, das jedoch global für alle Agenten zugreifbar sein muss.

Koordination von Viewern

Funktionalität zur Koordination mehrerer Viewer bei der Darstellung ein und desselben Teilbereichs wird gegebenenfalls in Form mehrerer Exemplare eines Managertyps benötigt. Die Anzahl der Exemplare ist abhängig von der Anzahl der Teilbereiche, die ein Informationssuchender gleichzeitig betrachten möchte. Die Hauptaufgaben dieser Art Manager sind vor allem:

- die Verwaltung des darzustellenden Teilbereichs als eine Menge von Informationscontainern
- die Verwaltung der aktuell selektierten Container aus dieser Menge
- die Benachrichtigung der angemeldeten Viewer bei Änderungen.

Darüber hinaus müssen diese Manager mit der zentralen Registratur zur Weitergabe und Aktualisierung farbiger Kennzeichnungen, zur Veröffentlichung aller verfügbaren Viewerdienste sowie für das Anfordern eines neuen Vieweragenten kooperieren.

Die beiden verschiedenen Arten von Managementaufgaben werden wie folgt getrennten Klassen übertragen:

- Das Management der Recheresitzung erfolgt durch einen Agenten vom Typ `SessionManager`.
- Das Management darzustellender Teilbereiche übernehmen mehrere Agenten vom Typ `SubsetManager`.

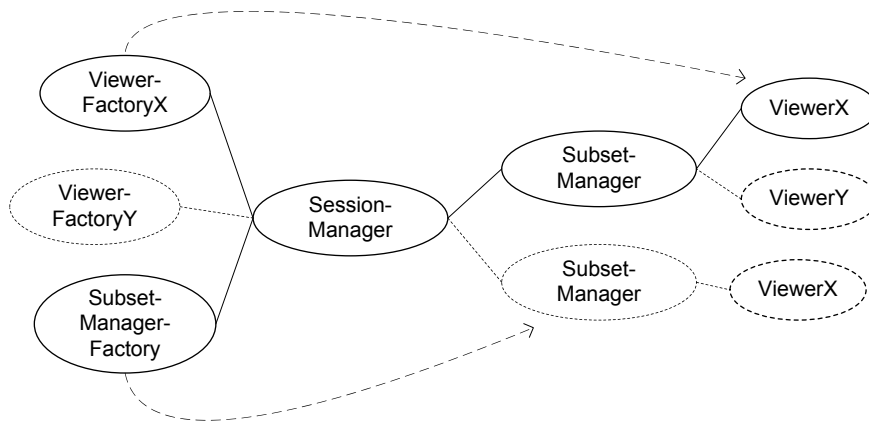


Abbildung 8.18.: Informale Darstellung der Gesamtstruktur des Frameworks

Daraus ergibt sich eine Gesamtstruktur des Frameworks wie im informalen Diagramm in Abbildung 8.18 dargestellt.

Das Registrieren aller `SubsetManager`-Agenten unmittelbar beim `SessionManager`-Agenten verhindert das Erzeugen unbegrenzt vieler Hierarchieebenen durch wiederholtes Starten von `SubsetManager`-Agenten für eine schrittweise immer weiter eingeschränkte Menge von Informationscontainern. Damit wird ein ineffizientes Weiterleiten von Aktualisierungsnachrichten über mehrere Hierarchieebenen vermieden.

Begrenzung der Hierarchie

Die beiden Klassendiagramme der Abbildungen 8.19 und 8.20 zeigen alle bisher diskutierten Agentenklassen einschließlich ihrer Attribute, Methoden und gültigen Nachrichten. Da `Viewer`- und `SubsetManager`-Agenten potentiell in beliebiger Anzahl benötigt werden, ist es für beide Klassen sinnvoll,

Statische Struktur

8. KOORDINATION VON RECHERCHEMODULEN

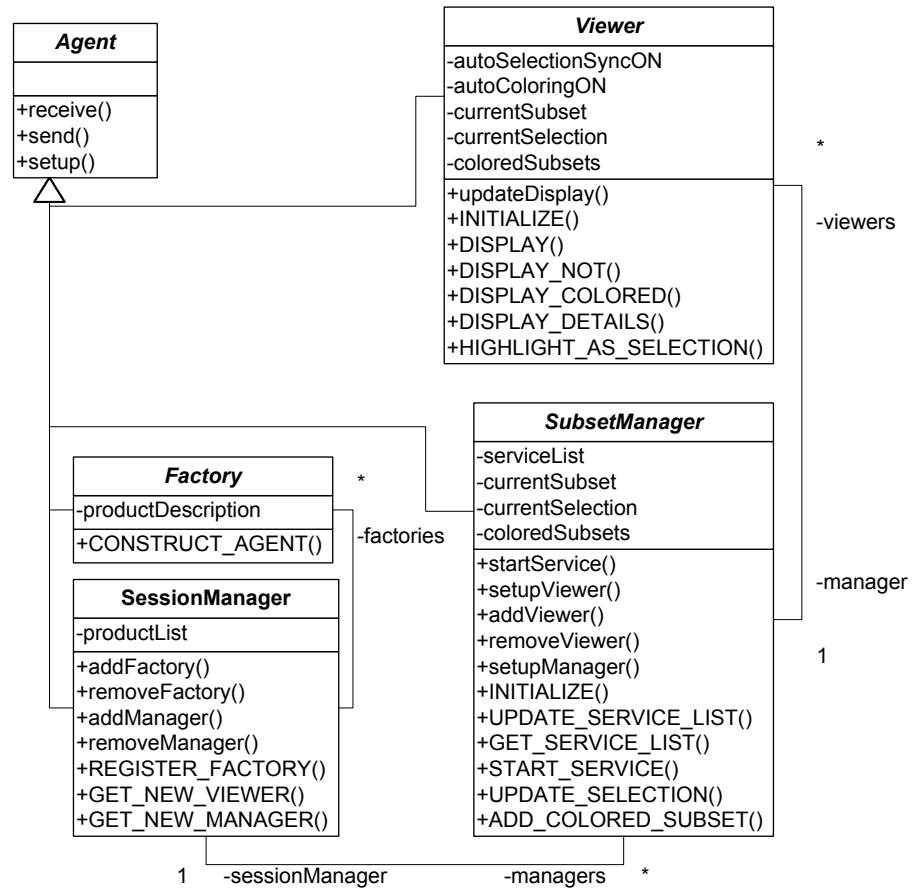


Abbildung 8.19.: Klassendiagramm zur Gesamtstruktur

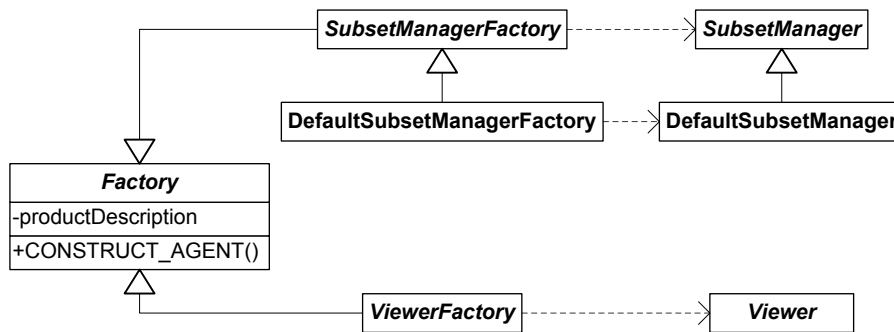


Abbildung 8.20.: Klassendiagramm: Fabriken und erzeugte Agententypen

Exemplare durch Fabrikklassen erzeugen zu lassen. Dies ermöglicht gleichzeitig die Erweiterung des Frameworks um **SubsetManager**-Implementierungen, die ggf. eine spezifische Logik für die Koordination von **Viewer**-agenten mitbringen. Die Klasse **SessionManager** beinhaltet die zentralen Dienste sowie die entsprechende Anwendungslogik für das Initialisieren und Koordinieren des Frameworks zur Laufzeit. Für diese Klasse ist es sinnvoll, eine Standardimplementierung vorzugeben, von der Exemplare erzeugt werden können. In diesem Fall wird keine abstrakte Klasse definiert. Alle anderen Klassen sind abstrakt und verlangen eine anwendungsspezifische Ableitung und Implementierung. Für die Klassen **SubsetManager** und **SubsetManagerFactory** sollten jedoch ebenso Standardimplementierungen in Form abgeleiteter konkreter Klassen bereitgestellt werden.

Der Nachrichtenaustausch zur Initialisierung der Informationsumgebung ist im Sequenzdiagramm in Abbildung 8.21 dargestellt. Zuerst wird ein **SessionManager**-Agent erzeugt und gestartet. Dieser wartet auf die Anmeldung eines Fabrikagenten zur Erzeugung von **SubsetManager**-Agenten. Nach dessen Anmeldung fordert der **SessionManager**-Agent einen neuen **SubsetManager**-Agenten an, registriert diesen bei sich selbst und sendet eine Nachricht zur Initialisierung. Dabei wird der gesamte verfügbare Infor-

Interaktionen
Start der Informationsumgebung

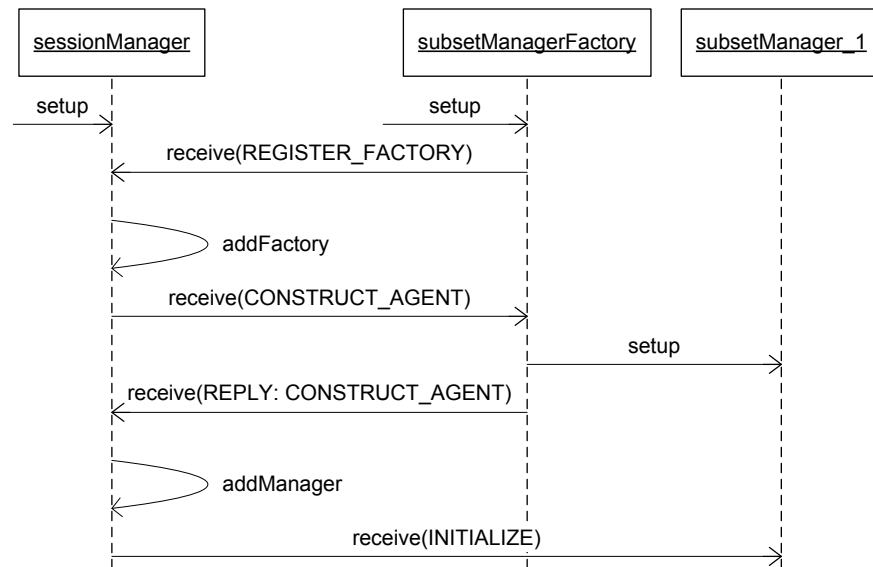


Abbildung 8.21.: Sequenzdiagramm: Starten einer Rechresitzung

mationsraum als aktuell darzustellender Teilbereich an den **SubsetManager**-Agenten übergeben.

*Registrieren eines
Fabrikagenten*

Abbildung 8.22 zeigt den Ablauf bei der Registrierung eines weiteren Fabrikagenten und dem Erzeugen eines neuen Vieweragenten. Die Anforderung eines neuen Vieweragenten erfolgt in diesem Fall aufgrund einer direkten Benutzerinteraktion mit dem aktiven **SubsetManager**-Agenten wie in Schritt 2 des Szenarios (vgl. S. 194) beschrieben. Das Registrieren des Fabrikagenten und das Anfordern des Vieweragenten können zeitlich getrennt erfolgen, wobei jedoch die Verfügbarkeit der betreffenden Fabrik Voraussetzung für das Erzeugen eines Viewers ist.

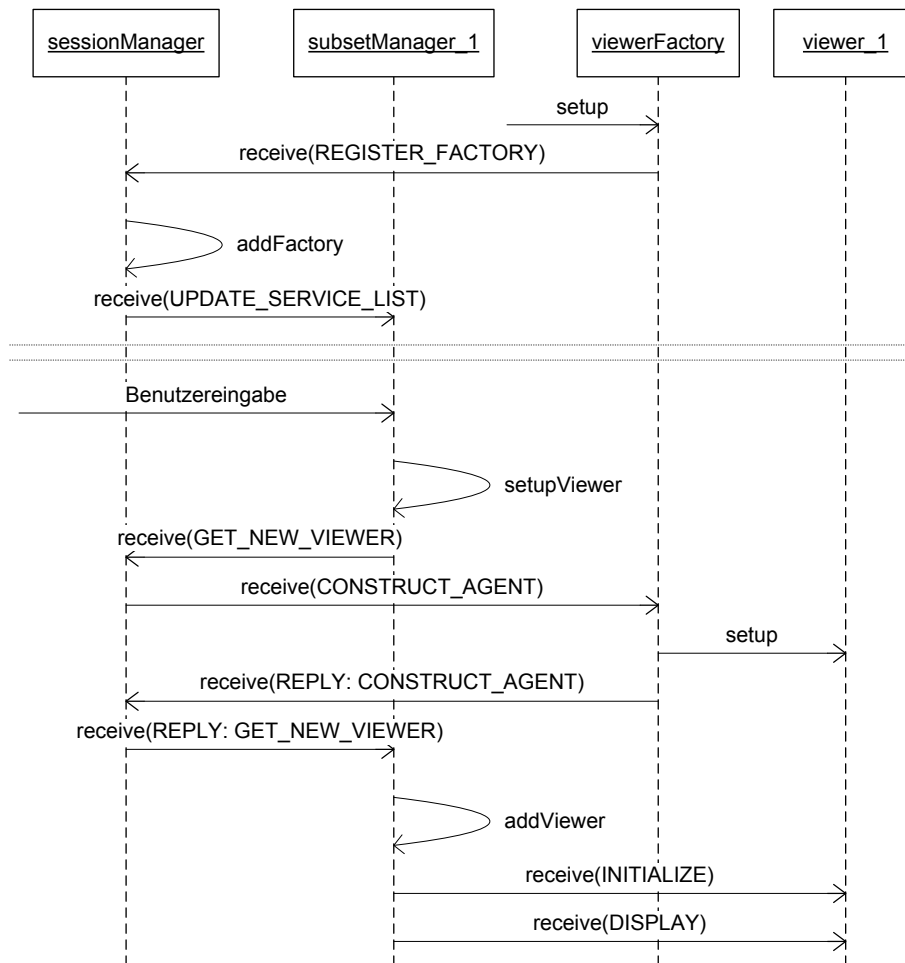


Abbildung 8.22.: Sequenzdiagramm: Anfordern eines neuen Vieweragenten

8.3.4. ZENTRALER ZUGRIFF AUF DIE ERSCHLIESSUNGSSTRUKTUR UND DEN MODELLVERBUND

Problem Die bisher diskutierten Lösungsansätze betreffen ausschließlich die Erzeugung, Interaktion und Koordination der an einer Recheresitzung beteiligten Viewer und den erforderlichen Manageragenten. Grundvoraussetzung für die Aktivität aller Viewer ist jedoch der Zugriff auf den Informationsraum. Da prinzipiell alle Viewer, aber auch alle **SubsetManager**-Agenten Zugang zur Erschließungsstruktur und gegebenenfalls auch zum Modellverbund haben müssen, ist die Bereitstellung an zentraler, für alle beteiligten Agenten zugreifbarer Stelle erforderlich. Abbildung 8.23 zeigt die angestrebte Konstellation.

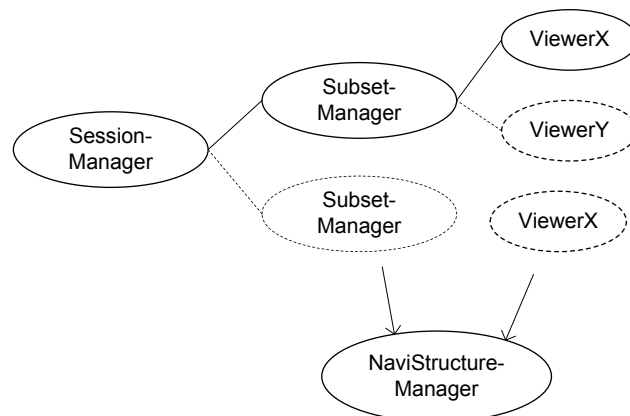


Abbildung 8.23.: Informale Darstellung zum zentralen Zugriff auf den Informationsraum

Relevantes Muster Dies kann mit Hilfe des *Singelton*-Musters (siehe [Gamma u. a., 2004, S. 157ff.] sowie Anhang A) gelöst werden. Alternative Muster für diese Aufgabenstellung sind nicht bekannt.

Unter Verwendung des *Singleton*-Musters kann eine Klasse `NaviStructureManager` auf zwei verschiedene Weisen in das Framework integriert werden:

*Diskussion
möglicher Ansätze*

- als *Singleton*-Klasse, auf deren einziges Exemplar alle Agenten bei Bedarf über die `exemplar()`-Methode eine Referenz anfordern können, oder
- als selbständiger Agententyp, der mit einer Referenz auf das Exemplar einer *Singleton*-Klasse zur Verwaltung der Erschließungsstruktur konfiguriert wird.

Die Möglichkeiten zur Implementierung von `NaviStructureManager` als Agent und gleichzeitig *Singleton*-Klasse hängen von der tatsächlichen Implementierung des Frameworks ab. Wird dabei zur Realisierung der Agentenaspekte auf bereits bestehende Frameworks wie das Java Agent Development Framework⁴ (JADE) zurückgegriffen, müssen die von diesem Framework bereitgestellten Mechanismen zur Erzeugung neuer Agenten berücksichtigt werden. JADE sieht beispielsweise die Erzeugung eines neuen Agenten durch die Erzeugung eines neuen Exemplars einer Klasse `AgentController` vor. Dies erfolgt durch einen parametrisierten Aufruf an ein Exemplar von `ContainerController`, wobei der Name des neuen Agenten sowie Name und Pfad der auszuprägenden Agentenklasse übergeben werden. Eine Manipulation dieses Mechanismus ist mit einem Eingriff in die interne Struktur und Interaktion des JADE-Frameworks verbunden.

Singleton-Agenten

Im Rahmen des bisherigen Konzepts ist der zweite Weg flexibler, da die übrigen Agenten keine Annahmen bezüglich der Klasse, die die Erschließungsstruktur verwaltet, treffen müssen. Der `NaviStructureManager`-Agent entkoppelt die Klasse von den übrigen Komponenten des Frameworks und ermöglicht so ein einfaches Austauschen, beispielsweise bei einer Ände-

*Agent mit Referenz
auf Singleton*

⁴ <http://jade.tilab.com>

rung der verwendeten Datenverwaltungstechnologie. Exemplare einer Agen-
 tenklasse `NaviStructureManager` erhalten somit jeweils eine Referenz auf
 das einzige Exemplar von `NaviStructure`.

Statische Struktur Die beteiligten Klassen sind im Diagramm in Abbildung 8.24 dargestellt.

Registrierung Für die Sicherstellung der globalen Verfügbarkeit der vom `NaviStructu-
 reManager`-Agenten angebotenen Dienste erfolgt die Registrierung einmalig
 beim `SessionManager`-Agenten.

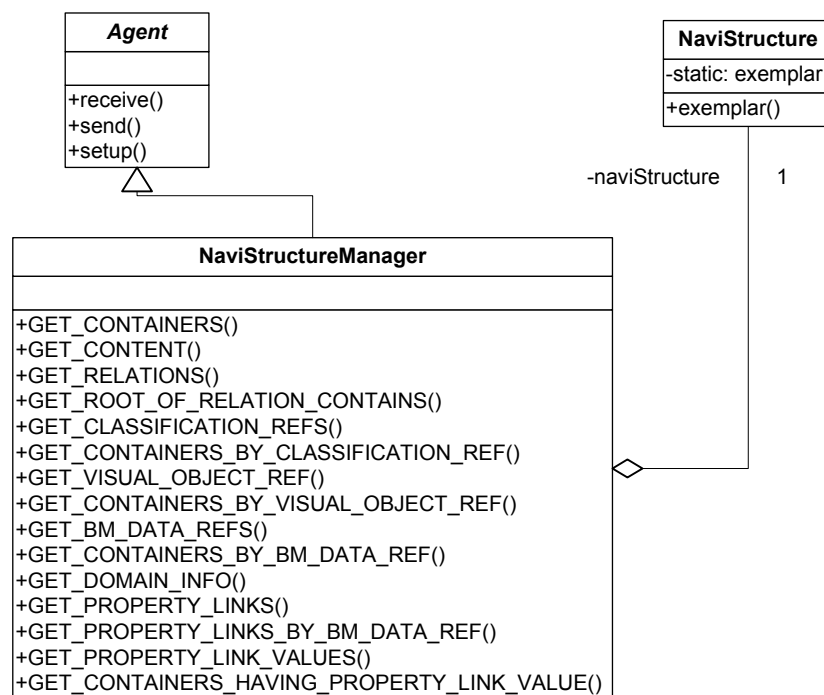


Abbildung 8.24.: Klassendiagramm: Zugriff auf die Erschließungsstruktur über die Klassen `NaviStructure` und `NaviStructureManager`

Zugriffsschnittstelle Korrespondierend zu den für `NaviStructureManager`-Agenten gültigen
 Nachrichten stellt die Klasse `NaviStructure` entsprechende Methoden zum
 Zugriff auf einzelne Bestandteile der Erschließungsstruktur zur Verfügung

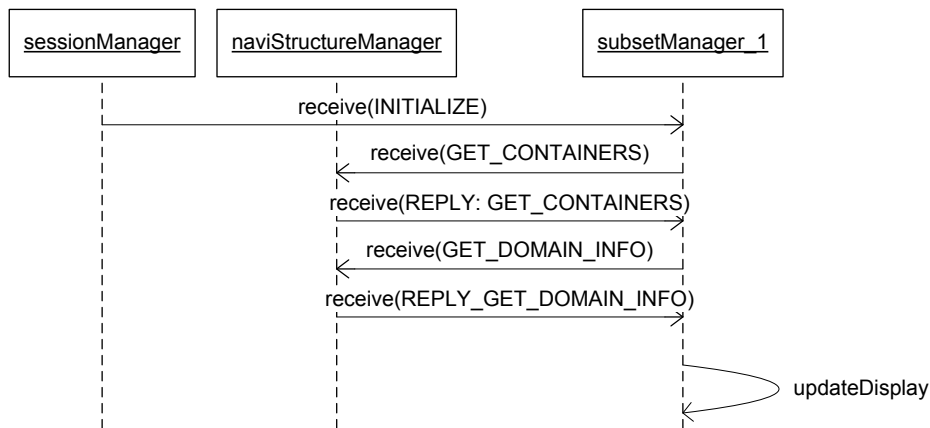


Abbildung 8.25.: Sequenzdiagramm: Aufbau der Übersicht

(vgl. Anhang B).

Abbildung 8.25 setzt das Sequenzdiagramm aus Abbildung 8.21 fort und zeigt den Nachrichtenaustausch zwischen `NaviStructureManager`-Agent und einem `SubsetManager`-Agenten beim Starten und Initialisieren der Informationsumgebung (Schritt 1 des Szenarios, S. 194). Nach Abschluss dieser Interaktionen kann der `SubsetManager`-Agent die Übersicht der Recheresitzung erstellen (Aufruf der Methode `updateDisplay()`). Abbildung 8.26 zeigt in Fortsetzung des Sequenzdiagramms aus Abbildung 8.22 die Kommunikation der Agenten zur Ermittlung der darzustellenden Inhalte in Schritt 2 des Szenarios (vgl. S. 194).

Interaktionen

8.3.5. ALPHANUMERISCHE AUSWERTUNG VON REFERENZKNOTEN

In Schritt 7 des Szenarios (vgl. Seite 199) wurde eine Detailsicht für ein einzelnes `InfContainer`-Exemplar geöffnet. Zur Generierung einer solchen Übersicht müssen Informationen aus den einzelnen, einem Container zuge-

Alphanumerische Informationen über Referenzknoten

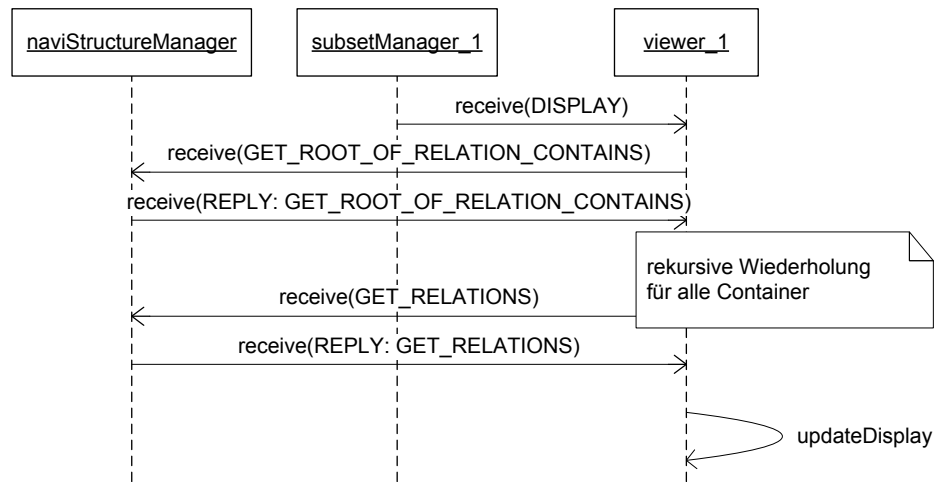


Abbildung 8.26.: Sequenzdiagramm: Ermitteln darzustellender Inhalte

ordneten Referenzknoten extrahiert und in alphanumerischer Form dargestellt werden können. Dafür werden Methoden benötigt, die einzelne konkrete Referenzknotenklassen verarbeiten und die benötigten Informationen als Zeichenketten zurückliefern können.

Prozessoren Die Implementierung dieser Methoden hängt jedoch jeweils von den konkreten Unterklassen einzelner abstrakter Referenzknotenklassen ab und kann so nicht durch das Framework realisiert werden. Innerhalb des Frameworks werden stattdessen für die einzelnen Referenzknotenklassen der Erschließungsstruktur abstrakte Prozessoren mit den entsprechenden Schnittstellen definiert (siehe Klassendiagramm in Abbildung 8.27). Diese müssen jeweils korrespondierend zu einer konkreten Referenzknotenklasse abgeleitet und implementiert werden.

Zuordnung zwischen Prozessoren und Referenzknoten Um die Zuordnung eines konkret benötigten Prozessors zu Exemplaren einer konkreten Referenzknotenklasse und dessen Ermittlung zur Laufzeit sicherzustellen, besitzt jede Referenzknotenklasse ein statisches Attribut `handlerService`, das den Namen des benötigten Prozessorendiens-

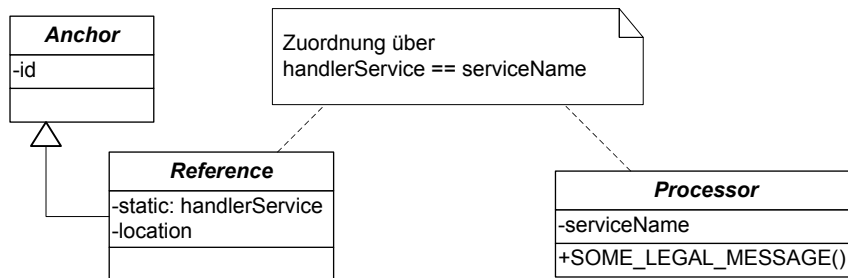


Abbildung 8.27.: Klassendiagramm: Zuordnung von Prozessoren zu Referenzknotenklassen

tes enthält. Die Prozessordienste werden, wie auf Seite 223 beschrieben, durch die Fabrikagenten realisiert. Bei der Anmeldung eines Fabrikagenten werden die Dienste unter dem übergebenen Namen in den Katalog des `SessionManager`-Agenten aufgenommen.

Während des Generierens einer Detailsicht geht ein Vieweragent Schritt für Schritt die dem aktuell anzuzeigenden Informationscontainer zugeordneten Referenzknotenexemplare durch und ermittelt anhand des Wertes von `handlerService` in Kooperation mit dem `SessionManager`-Agenten den zugehörigen Processoragenten. Von letzterem werden per Nachrichtenaustausch die benötigten Informationen zum jeweiligen Referenzknotenexemplar angefordert. Ein Überblick zu den benötigten Prozessorentypen, den zu ermittelnden Informationen bzw. gültigen Nachrichten für die einzelnen Referenzknotenklassen kann Anhang C entnommen werden.

Kommunikation der Agenten

Ebenso wie Prozessoren beziehen sich auch konkrete Viewerklassen stets auf die Auswertung einer bestimmten konkreten Referenzknotenklasse. Die Zuordnung folgt demselben Prinzip wie für Prozessoren. Viewerdienste werden jeweils unter einen Namen registriert, der dem Wert von `handlerService` in der vom Viewer verarbeitbaren Referenzknotenklasse entspricht. Die Unterscheidbarkeit der Katalogeinträge bleibt anhand des Diensttyps

Zuordnung zwischen Viewern und Referenzknoten

gegeben. Das Diagramm in Abbildung 8.28 zeigt exemplarisch die benötigten konkreten Klassen zur Auswertung und Darstellung eines speziellen Typs von Referenzknoten. Dieser verweist auf Konzepte, die in einer formal repräsentierten Ontologie definiert wurden, wie sie in Abschnitt 7.3.2 beschrieben wurde.

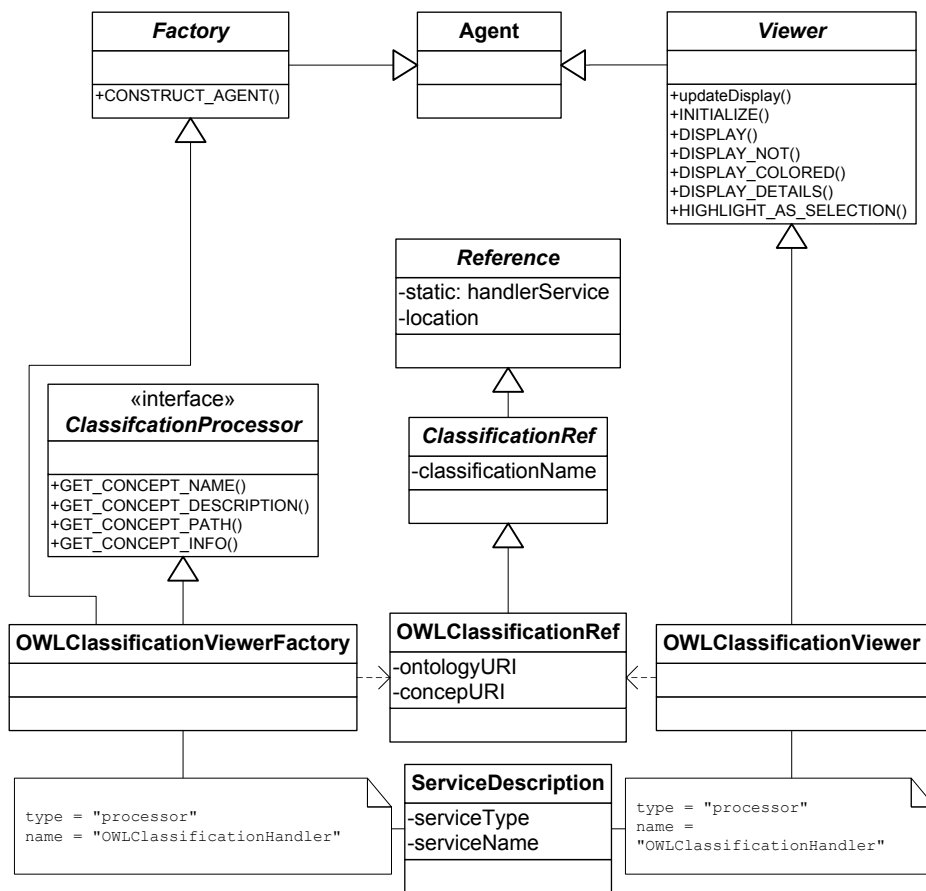


Abbildung 8.28.: Klassendiagramm: Zuordnung von Viewerdiensten zu Referenzknotenklassen (exemplarisch)

8.4. IMPLEMENTIERUNG DES RECHERCHEFRAMEWORKS

Ein wesentliches Merkmal des vorgeschlagenen Systems ist die lose Kopplung der einzelnen Komponenten (Agenten) und deren Kommunikation durch asynchronen Austausch von Nachrichten. Zu diesem Zweck wird eine Infrastruktur benötigt, die allen Komponenten einen zentralen Dienst für den Nachrichtenaustausch bereitstellt, so dass sich die Komponenten unabhängig voneinander bei diesem Dienst registrieren, dort Nachrichten für andere Komponenten ablegen und eigene Nachrichten abrufen können. Darüber hinaus wird eine Registratur benötigt, bei der sich Komponenten nach dem Start anmelden und die von ihnen angebotenen Dienste registrieren, so dass sie von anderen Komponenten gefunden werden können.

Anforderungen

Eine solche Infrastruktur wird u. a. durch das Java Agent Development Framework (JADE)⁵ bereitgestellt. Unter Verwendung dieses frei verfügbaren Frameworks wurde die Umsetzbarkeit des vorgeschlagenen Rechercheframeworks in seiner Grundstruktur überprüft (siehe Anhang D.4). Dazu zählten insbesondere die Kommunikation der einzelnen Agenten beim Aufbau der Hierarchie, beim Anmelden eines neuen Fabrikagenten und beim Anfordern und Erzeugen eines neuen Vieweragenten. Eine vollständige Implementierung des Frameworks als Nutzerschnittstelle einschließlich verschiedener Viewermodule bleibt jedoch weiterführenden Arbeiten vorbehalten.

*Verfügbare
Technologie*

Ausgehend von dem zu Beginn des Kapitels beschriebenen Szenario (vgl. S. 192ff.) wird im Folgenden der Aufbau der entsprechenden Agentenstruktur für die Schritte 1 und 3 des Szenarios erläutert.

⁵ siehe <http://jade.tilab.com/>

Initialisierung **Schritt 1 des Szenarios: Starten der Informationsumgebung:** Folgende Arbeitsschritte sind systemintern auszuführen:

- Arbeitsschritte*
1. Starten eines Agenten vom Typ `SessionManager` (*sessMan*)
 2. Starten eines Agenten vom Typ `DefaultSubsetManagerFactory` (*defSSMFact*) und Anmelden desselben bei *sessMan*
 3. Erzeugen eines Agenten vom Typ `DefaultSubsetManager` (*stdSSMan*) und Anmelden bei *sessMan*.

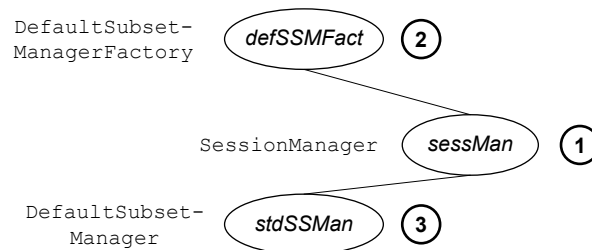


Abbildung 8.29.: Agentenhierarchie nach dem Start der Informationsumgebung

Status der Hierarchie Abbildung 8.29 zeigt schematisch den Zustand der Agentenhierarchie bis zu diesem Zeitpunkt, wobei die Reihenfolge der Aktivierung entsprechend der Arbeitsschritte gekennzeichnet ist.

Nutzersicht Abbildung 8.30 zeigt rechts die von JADE bereitgestellte grafische Oberfläche zum Management aktuell aktiver Agenten. Die drei aktivierten Agenten werden im unteren Teil der Baumstruktur in einem eigenen Container dargestellt. Auf der linken Seite ist die grafische Oberfläche des `SessionManager`-Agenten *sessMan* dargestellt.

Start eines Viewers **Schritt 3 des Szenarios: Starten des 3D-Viewers** für das 2. Obergeschoss: Bis zu diesem Zeitpunkt wurde ein weiterer Agent (*aggrView*) aktiviert und *stdSSman* zugeordnet. Dieser realisiert die Darstellung der räumliche Aggregationsstruktur des Gebäudes. Der Nutzer hat darin das 2.

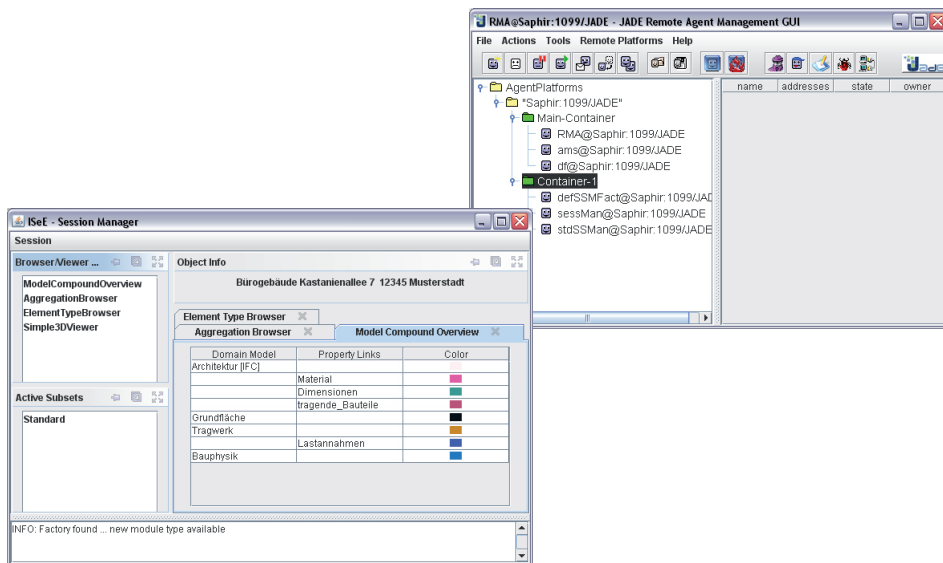


Abbildung 8.30.: Session Manager und Jade Remote Agent Management GUI

Obergeschoss gewählt und für dieses die Darstellung in einem 3D-Viewer angefordert. Ein entsprechender Fabrikagent (*simple3DViewFact*) zur Erzeugung von 3D-Viewer-Exemplaren wurde bereits bei *sessMan* angemeldet (0). Systemintern sind daraufhin folgende Arbeitsschritte auszuführen:

1. Anfordern, Erzeugen und Anmelden eines neuen Agenten vom Typ *DefaultSubsetManager* (*ssMan2*) *Arbeitsschritte*
2. Übergabe aller Informationscontainer, die zum 2. Obergeschoss gehören an *ssMan2* und Aufforderung zur Darstellung als 3D-Szene
3. Anfordern, Erzeugen und Anmelden eines neuen Agenten vom Typ *Simple3DViewer* (*simple3DView*)
4. Übergabe aller *ssMan2* bekannten Informationscontainer an *simple3DView* und Aufforderung zur Darstellung.

Status der Hierarchie Abbildung 8.31 zeigt wiederum den Zustand der Agentenhierarchie nach dem Start von *simple3DView*.

Nutzersicht In Abbildung 8.32 sind die einzelnen Schritte anhand der grafischen Benutzeroberflächen der einzelnen Agenten dargestellt.

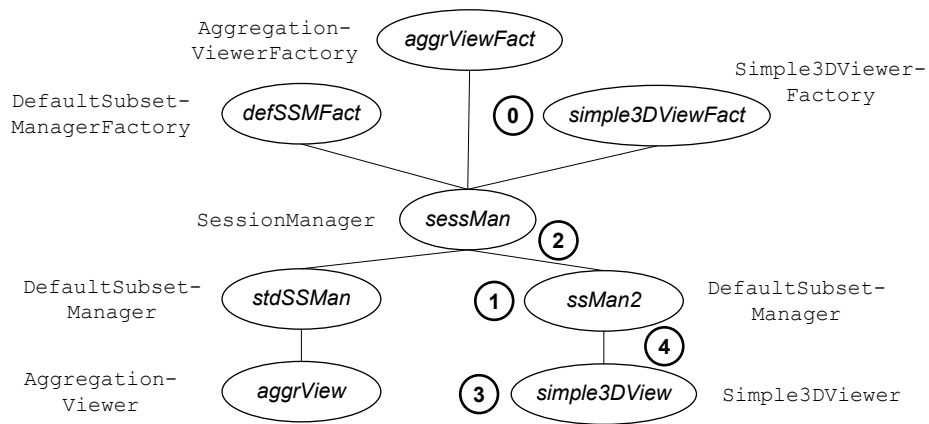


Abbildung 8.31.: Agentenhierarchie nach dem Start des 3D-Viewers

8.4. IMPLEMENTIERUNG DES RECHERCHEFRAMEWORKS

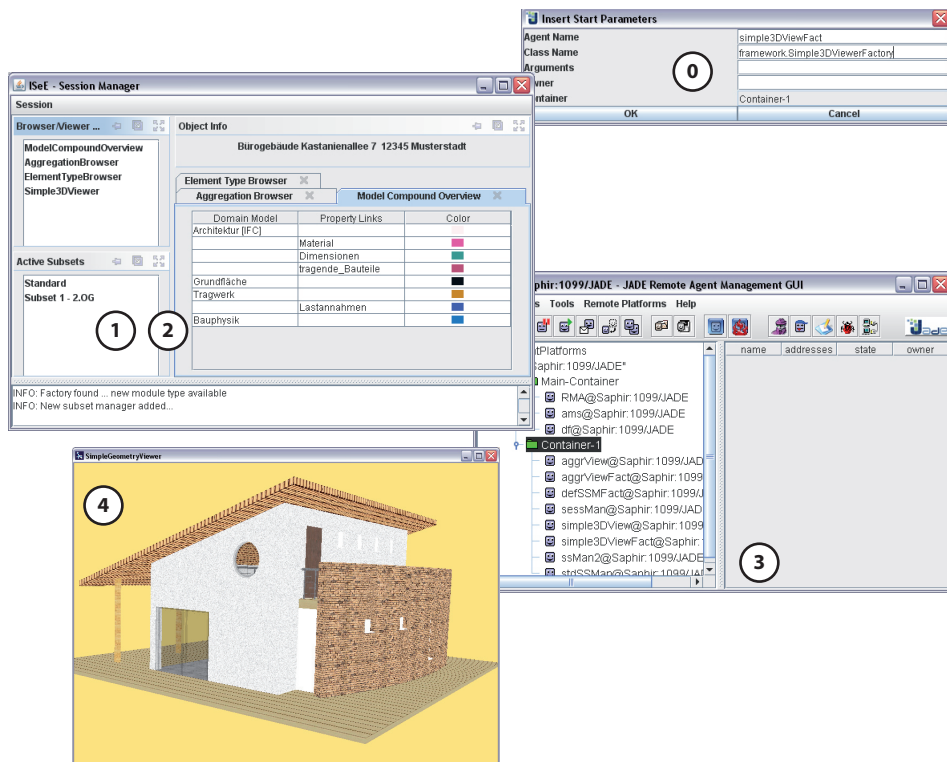


Abbildung 8.32.: Benutzeroberflächen einzelner aktiver Agenten

8.5. DISKUSSION

Zwischenergebnis In diesem Kapitel wurde auf der Basis der Objektorientierung ein System beschrieben, das die Nutzerschnittstelle einer digitalen Bauwerksdokumentation realisiert. Die vorgeschlagene Systemstruktur basiert auf der hierarchischen Organisation lose gekoppelter Komponenten verschiedenen Typs:

- *Viewer* stellen – als Recherchemodule – Präsentations- und Interaktionsfunktionalität bereit.
- *Fabriken* dienen der Integration neuer Viewerklassen sowie der Erzeugung entsprechender Exemplare.
- *Manager* koordinieren die zur Laufzeit aktiven Fabriken und Viewer.

Die beteiligten Komponententypen wurden in Form abstrakter Klassen spezifiziert, für die jeweils die erforderlichen Schnittstellen und Interaktionsmuster festgelegt wurden. Das so entstehende Grundgerüst (*Framework*) für die Nutzerschnittstelle erlaubt das flexible Hinzufügen beliebiger Recherchemodule durch Ableitung und Implementierung der vorgegebenen abstrakten Klassen. Das Framework wurde in Teilen exemplarisch realisiert, wobei kritische Aspekte hinsichtlich ihrer Umsetzbarkeit überprüft wurden (dazu siehe Anhang D).

Präsentations- und Interaktionsfunktionalität Die Implementierung einzelner Recherchemodule stand dabei nicht im Vordergrund. Neben einer an spezifische Anforderungen angepassten Neuimplementierung von Präsentations- und Interaktionsfunktionalität ist diesbezüglich die Integration bestehender Viewer – in Korrespondenz zu den jeweils verwendeten konkreten Referenzknotenklassen und externen Inhalten – denkbar. Dafür können bestehende Viewerapplikationen durch einen so genannten *Wrapper*-Agenten umhüllt werden. Dieser muss einerseits die abstrakte Klasse `Viewer` ableiten und implementieren, andererseits die Kommunikation mit der zu integrierenden Viewerapplikation über deren Schnitt-

stelle realisieren. Dies wurde jedoch nicht weiter untersucht.

Für die Entwicklung des Frameworks wurde primär auf Funktionalität zur Unterstützung explorativer Suche fokussiert. Um analytische Suchstrategien ebenso unterstützen zu können, wird ergänzende Funktionalität zur Handhabung formaler Anfragen benötigt. Dabei müssen zwei Aspekte berücksichtigt werden:

*Unterstützung
analytischer Suche*

- Funktionalität, die dem Informationssuchenden die Definition formaler Anfragen ermöglicht, sowie
- Funktionalität, die systemintern die Abarbeitung formaler Anfragen übernimmt (Realisierung des Matching-Paradigmas).

Beim ersten Aspekt handelt es sich um spezifische Formen von Präsentations- und Interaktionsfunktionalität. Die Realisierung fällt damit spezifischen Typen von Recherchemodulen zu. Dabei sollte insbesondere Unterstützung für eine interaktive Spezifikation von Anfragen – auch ohne Kenntnis formaler Anfragesprachen – berücksichtigt werden.

Anfragedefinition

Der zweite Aspekt stellt eine Basisfunktionalität des Systems dar, die – ebenso wie der Zugriff auf Erschließungsstruktur und Modellverbund – an zentraler Stelle für beliebige Module verfügbar sein muss. Diesbezüglich bietet sich eine Erweiterung der Klassen `NaviStructure` und `NaviStructureManager` an. Auf diese Weise wäre eine analytische Auswertung der ES-Daten möglich. Anfragefunktionalität zur Auswertung des Modellverbundes bzw. einzelner Domänenmodelle muss jeweils korrespondierend zu den Schemata sowie zur verwendeten Technologie der Modellverwaltung erfolgen. Diesbezüglich bietet sich die Realisierung und Integration in das Framework in Form einzelner, jeweils modellspezifischer Komponenten (Agenten) an. Entsprechende Untersuchungen fanden in Ansätzen in parallelen Arbeiten statt [Riedel u. Wender, 2007], eine Vertiefung bleibt jedoch weiterführenden Arbeiten vorbehalten.

Anfragebearbeitung

9. FAZIT

Aufgrund der Komplexität des Planungsgegenstandes haben sich in der kooperativen Bauplanung Ansätze zur Verwaltung bauwerksbezogener Informationen in Form eines variablen Verbundes fachspezifischer objektorientierter Modelle etabliert. Die Konstellation eines solchen Modellverbundes ergibt sich inhaltlich aus der differenzierten Sicht der verschiedenen fachlich Beteiligten auf den Planungsgegenstand sowie aufgrund der mangelnden Beherrschbarkeit einer diese Sichten vollständig integrierenden Datenstruktur in Form eines konzeptuellen Schemas. Eine so organisierte digitale Repräsentation eines Bauwerks eignet sich aufgrund ihrer Flexibilität für die lebensphasenübergreifende Fortschreibung in Form einer „digitalen Bauakte“ bzw. Bauwerksdokumentation. Diese wiederum stellt eine wichtige Grundlage für die Planung von Instandhaltungs-, Modernisierungs-, Umbau- oder Erweiterungsmaßnahmen am betreffenden Bauwerk dar.

Gegenstand

Aufgrund der strukturellen Zerlegung in einzelne fachspezifische Modelle ist ein solcher Modellverbund gekennzeichnet durch das Vorhandensein verteilter fachspezifischer Repräsentationen jeweils eines Realweltobjektes, d. h. eines real existierenden oder geplanten Elementes eines Bauwerks wie beispielsweise eines Raumes oder Bauteils. Bestehende Ansätze zur Informationsbereitstellung besitzen ein Anforderungsniveau, das für die Akteure insbesondere in frühen Planungsphasen nicht akzeptabel bzw. nicht beherrschbar ist, da eine fachübergreifende Sicht auf einem sich an den Denkweisen der Informationssuchenden orientierenden und von den informationstechnischen Interna abstrahierenden Niveau nicht gegeben ist.

Problem

- Identifikatoren als zentrale Einstiegspunkte* Durch die Verknüpfung aller fachspezifischen Repräsentationen mit einem eindeutigen Identifikator, der jeweils ein Realweltobjekt vertritt, können zentrale Einstiegspunkte bereitgestellt werden, die die Informationssuche im Datenbestand des Modellverbundes erheblich erleichtern. Mit Hilfe der Identifikatoren kann durch explizite Zuordnung von Elementtypen, topologischen Beziehungen und Visualisierungsinformationen die baulich-räumliche Struktur des konkreten Bauwerks repräsentiert und als fachübergreifendes Ordnungssystem für die Suche verfügbar gemacht werden. Die Identifikatoren bilden dabei jeweils den Kern eines erweiterbaren „Informationscontainers“ zu je einem Realweltobjekt, ohne dessen verteilte, fachspezifische Repräsentationen aufzulösen.
- Präsentation und Interaktion* Auf der Basis des Informationscontainer-Konzeptes ist eine modulare, flexibel erweiterbare Nutzerschnittstelle realisierbar, die die baulich-räumliche Struktur den Informationssuchenden in Form interaktiver Präsentationen zur Verfügung stellt. Auf diese Weise wird die Interaktion mit dem digital repräsentierten Informationsraum, bestehend aus Ordnungssystem und Modellverbund, als Voraussetzung für die Informationssuche ermöglicht.
- Systemphilosophie* Im Rahmen der Arbeit wurde ein Gesamtkonzept für eine entsprechende Informationsumgebung entwickelt, wobei die Ebene des Modellverbundes als gegeben vorausgesetzt wurde. Die ergänzenden Komponenten *Erschließungsstruktur* und *Nutzerschnittstelle* wurden in Form eines konzeptuellen Schemas bzw. objektorientierten Systems spezifiziert. Beides wurde unter Verwendung der UML auf implementierungsunabhängiger Ebene beschrieben. Das Systemkonzept wurde in Teilen prototypisch realisiert, wobei jeweils die Umsetzbarkeit einzelner, als kritisch erachteter Aspekte überprüft wurde (siehe Anhang D).

9.1. WERTUNG

Das vorliegende Systemkonzept stellt eine funktionale Lösung für das betrachtete Problem der Informationssuche in einer digitalen Bauwerksdokumentation dar, wobei bei der Lösungsfindung primär auf den Kontext der frühen Planungsphasen fokussiert wurde. Die erforderlichen Komponenten wurden hinsichtlich ihrer Verantwortlichkeiten, Funktionalitäten, Beziehungen und Interaktionen beschrieben. Ausgehend von dieser funktionalen Lösung sind mehrere spezifische Lösungen möglich, d. h. verschiedene Varianten der Implementierung gegeben, die hier jedoch nicht abschließend behandelt werden sollten.

Funktionale Lösung

Im Zusammenhang mit der Wahl der zur Implementierung verwendeten Technologien ist eine weitere Detaillierung des Konzeptes und ggf. eine Optimierung in Bezug auf Performance-Aspekte erforderlich. Davon bleibt jedoch das Prinzip der funktionalen Lösung unberührt. Insbesondere durch den Rückgriff auf objektorientierte Entwurfsmuster als Essenz erfolgreicher praktischer Lösungen kann davon ausgegangen werden, dass eine vollständige Umsetzung des Systems ohne wesentliche strukturelle Modifikationen möglich ist.

Spezifische Lösung

Bezogen auf die Konkretisierungsstufen der Bauplanung ist das vorliegende Systemkonzept mit der Detaillierungsstufe der Genehmigungsplanung gleichzusetzen. Auf der Basis der formalisierten implementierungsunabhängigen Systembeschreibung sollte sowohl eine weiterführende Detaillierung und Implementierung als auch das Nachvollziehen und Wiederverwenden des Systementwurfs oder Teilen davon für ähnliche Problemstellungen möglich sein.

Einordnung der Ergebnisse

Den Ausgangspunkt für die Formulierung des vorgeschlagenen Systemkonzeptes bildeten explizit die frühen Planungsphasen, d. h. die Phasen der Konzeptfindung für ein Projekt im Kontext bestehender Bausubstanz.

Betrachteter Kontext

Vor diesem Hintergrund stellten Gebäude zunächst den primär fokussierten Bauwerkstyp dar. Darüber hinaus wurde in erster Linie auf die Unterstützung nicht vorhersehbarer Planungstätigkeiten mit einem hohen Anteil kreativer, nicht formalisierbarer Aktivitäten orientiert. Vor dem Hintergrund dieser einschränkenden Annahmen sind die Übertragbarkeit des vorgeschlagenen Lösungsansatzes sowie ggf. erforderliche Anpassungen oder Erweiterungen zu diskutieren, insbesondere hinsichtlich der Betrachtung

- weiterer (d. h. fortgeschrittener) Planungsphasen,
- weiterer Phasen des Bauwerkslebenszyklus sowie
- weiterer Bauwerkstypen.

Informationscontainer als flexibles Strukturprinzip Das Konzept der Informationscontainer, das eine eindeutige Repräsentation einzelner Realweltobjekte (=Bauwerkselemente) durch Identifikatoren und die Verknüpfung unterschiedlicher, das Realweltobjekt beschreibender Informationen mit den Identifikatoren vorsieht, ist nicht an die Raum-Bauteil-Struktur von Gebäuden gebunden. Die Identität, Bedeutung und Beziehungsstruktur der Realweltobjekte wird außerhalb der Informationscontainer in Form extern repräsentierter, durch die Container lediglich referenzierter Informationen beschrieben. Bezüglich der Art und Repräsentationsform dieser extern verwalteten Inhalte bestehen keine Restriktionen. Die Anwendbarkeit des Prinzips ist somit für beliebige Bauwerkstypen gegeben – sofern diese eine logische Gliederung in einzelne Elemente aufweisen, denen jeweils Informationscontainer zugeordnet werden können.

Kreative Tätigkeiten vers. Routineaufgaben Im Hinblick auf die Unterstützung kreativer Tätigkeiten beim Finden des Planungskonzeptes wurden Anschaulichkeit, Interaktivität sowie das Einladen des Nutzers zu informalen Vorgehensweisen als wesentliche Eigenschaften der Benutzerschnittstelle gefordert. In fortgeschrittenen Planungsphasen, aber auch im Tätigkeitsfeld anderer Fachdisziplinen sind jedoch oft Routineaufgaben auszuführen, für die systematische Vorgehensweisen

bzw. formalisierbare Lösungswege existieren. Auf der Basis der vorgeschlagenen modularen Struktur der Schnittstelle ist eine sehr flexible Realisierung und Integration unterschiedlicher Präsentations- und Interaktionstechniken möglich. Letztlich hängt es jedoch von der Konzeption und Implementierung der einzelnen Recherchemodule ab, inwieweit explorative oder analytische Vorgehensweisen bei der Suche unterstützt werden. Somit ist es möglich, für wiederkehrende Routineaufgaben, deren Informationsbedarf ganz oder teilweise vorformuliert werden kann, spezielle Module zu entwickeln, die diesen Informationsbedarf automatisiert oder teilautomatisiert unterstützen. In Abhängigkeit von den im konkreten Fall realisierten und verwendeten Modulen bleibt die Einsetzbarkeit der Informationsumgebung nicht auf die frühen Planungsphasen beschränkt.

*Verwendete Module
bestimmen
Einsatzbereich*

Während der Bauwerksnutzung werden im Rahmen des Facility Managements ebenfalls bauwerksbezogene Informationen benötigt, die oft mit weiteren nutzungsspezifischen Informationen wie stofflichen, energetischen oder monetären Kenngrößen verknüpft und ausgewertet werden müssen. Aufgrund der Flexibilität bezüglich Datenhaltung sowie Nutzerschnittstelle bzw. Anwendungslogik zur Informationsbereitstellung kann von einer prinzipiellen Übertragbarkeit des Systemkonzeptes auf die Phase der Bauwerksnutzung ausgegangen werden. Dabei wird jedoch eine Anpassung – insbesondere durch die Konzeption und Realisierung entsprechender Module – an die spezifischen Belange des Facility Managements erforderlich sein.

9.2. AUSBLICK

Im Zuge einer vollständigen Implementierung des Gesamtsystems wird eine technische Detaillierung und ggf. Optimierung des Systemkonzeptes in Korrespondenz zu gewählten Implementierungstechnologien erforderlich sein. Aus dieser Perspektive ist nur bedingt ein wissenschaftlicher Erkenntnis-

*Weiterführung auf
verschiedenen Ebenen*

gewinn zu erwarten. Für eine weiterführende Auseinandersetzung mit der dargestellten Problematik auf konzeptioneller Ebene bieten sich jedoch verschiedene Ansatzpunkte.

„Instanziierung“ der
Erschließungsstruktur

Im Mittelpunkt der vorliegenden Arbeit stand die Entwicklung und Beschreibung eines konzeptuellen Schemas zur Verwaltung der Erschließungsstruktur einer digitalen Bauwerksdokumentation. Der Prozess der Ausprägung des Schemas, d. h. des Erzeugens einer konkreten Erschließungsstruktur, wurde dabei nicht betrachtet. Für die Verfügbarkeit und Nutzbarkeit der Bauwerksdokumentation in der hier beschriebenen Form ist jedoch die Zuordnung von Verantwortlichkeiten, Methoden und Werkzeugen für die Erschließung des Informationsraums und die fortlaufende Pflege des „erschlossenen Informationsraums“ erforderlich. Diesbezüglich können abschließend nur einige Vorschläge unterbreitet werden, deren Realisierbarkeit nicht zuletzt von der Entwicklung einer entsprechenden Infrastruktur von durch Planende handhabbaren Generator-, Editor- und Viewerwerkzeugen abhängig ist.

Aufbau der
Erschließungsstruktur
während des
Entwerfens

Die Freiheit bei der Zuordnung beschreibender Informationen zu einzelnen Informationscontainern ermöglicht darüber hinaus den flexiblen Umgang mit Informationen während des Entwerfens: Entwurfsobjekte – als zunächst gedachte, später herzustellende Realweltobjekte – werden jeweils durch ihren Identifikator repräsentiert. Mit dem Fortschreiten des Entwurfsprozesses können sukzessive beschreibende Informationen mit den Identifikatoren verknüpft und so die Beschreibung des Entwurfsobjektes präzisiert werden. Eine Zuordnung von *Identität*, *Bedeutung* und *Struktur* bzw. deren Wechsel ist jederzeit möglich und unabhängig von bereits existierenden beschreibenden Informationen.

Automatisierte
Datenüberführung

Die zur Erzeugung der Informationscontainer einschließlich der Beschreibung ihrer Identität, Bedeutung und Struktur benötigten Informationen sind meist in einzelnen fachspezifischen Modellen bereits vorhanden. Als

wesentliche Informationsquelle sind hier architekturbezogene Modelle zu sehen. Zu überprüfen ist in diesem Zusammenhang die Übertragbarkeit bereits existierender Ansätze zur automatisierten Datenüberführung zwischen einzelnen Modellen [Willenbacher, 2002], mit deren Hilfe ein Aufbau der Ordnungsstruktur parallel zur Entwicklung eines architekturenspezifischen Modells denkbar ist.

Aus der Perspektive einer Informationsumgebung wurde zunächst nur von einem lesenden Zugriff auf den Informationsraum ausgegangen. Grundsätzlich jedoch sollen die verfügbaren Informationen als Grundlage für Planungsaktivitäten bereit stehen. Vor diesem Hintergrund ist eine Weiterführung des Ansatzes im Sinne einer Integration zwischen der Funktionalität der Informationsumgebung und der Funktionalität zur Bearbeitung der Daten erforderlich. Dabei sind einerseits bestehende Fachapplikationen einzubinden, da diese die notwendige Verarbeitungslogik zur Bearbeitung einzelner fachspezifischer Modelle bereitstellen. Andererseits muss fachübergreifend sowohl die Bearbeitung der Erschließungsstruktur-Daten als auch das Management kooperativer Planungsprozesse sichergestellt werden. Dies bedeutet unter anderem auch ein Herstellen von Querbezügen zu entsprechenden Arbeitsergebnissen [Rüppel, 2007].

Planungsaktivitäten

Insgesamt spielt für eine durchgängige Unterstützung des Planungsprozesses die Integration der frühen Phasen, in denen verstärkt kreative, nicht formalisierbare Tätigkeiten auszuführen sind, eine entscheidende Rolle. In diesen Phasen werden die für ein Projekt maßgeblichen Rahmenbedingungen in Form des Projektkonzeptes festgelegt. Gerade um Entwerfenden dabei die Möglichkeit für das Finden neuer, kreativer, innovativer Lösungen offen zu halten, sollten die bereitstehenden Werkzeuge die Offenheit des Lösungsprozesses nicht künstlich einschränken. Im Mittelpunkt der Werkzeugentwicklung sollten daher grundsätzlich die menschlichen Fähigkeiten stehen, die durch entwurfsunterstützende Werkzeuge gefördert und gegeben

Anspruch an entwurfsunterstützende Werkzeuge

nenfalls augmentiert werden. Die Verantwortung für den Entwurf – sowohl im Detail als auch in seiner Gesamtheit – obliegt ungeachtet des Fortschritts technischer Werkzeuge stets dem Entwerfenden. Ein intuitiver, spielerischer, anschaulicher Umgang mit dem Planungsmaterial schafft für den Entwerfenden die Grundlage, eben diese Verantwortung übernehmen zu können.

LITERATURVERZEICHNIS

- [Alexander u. a. 1995] ALEXANDER, C. ; ISHIKAWA, S. ; SILVERSTEIN, M. ; JACOBSON, M. ; KING, I. F. ; ANGEL, S.: *Eine Mustersprache*. Wien : Löcker Verlag, 1995
- [Anderson u. a. 1999] ANDERSON, K.M. ; DAVIS, H. ; GRØNBÆK, K. ; HAAKE, J.M. ; MILLARD, D. ; NÜRNBERG, P.J. ; REICH, S. ; WIIL, U.K.: Addressing Interoperability in Open Hypermedia: The Design of the Open Hypermedia Protocol. In: *New Review of Hypermedia and Multimedia* 5 (1999), S. 207–248
- [Anderson u. a. 1997] ANDERSON, K.M. ; TAYLOR, R.N. ; WHITEHEAD, E.J.: A Critique of the Open Hypermedia Protocol. In: *Journal of Digital Information (JoDi). Special Issue on Hypermedia Systems* 1 Issue 2 (1997), S. Article No. 5
- [Bakkeren u. Tolman 1995] BAKKEREN, W. J. C. ; TOLMAN, F. P.: Integrating structural synthesis and evaluation using product models. In: PAHL, P. J. (Hrsg.) ; WERNER, H. (Hrsg.): *Proceedings of the 6th International Conference on Computing in Civil & Building Engineering (ICCCBE)*. Berlin : A. A. Balkema, 1995
- [Balzert 2001] BALZERT, H.: *Lehrbuch der Software-Technik*. Heidelberg : Spektrum Akademischer Verlag, 2001
- [Balzert 2005] BALZERT, H.: *Lehrbuch der Objektmodellierung*. München : Spektrum Akademischer Verlag, 2005
- [Banyard u. a. 1995] BANYARD, P. ; CASSELLS, A. ; GREEN, P. ; HARTLAND, J.

; HAYES, N. ; REDDY, P.: *Einführung in die Kognitionspsychologie*. München u.a. : Ernst Reinhard Verlag, 1995

[BARBi Project] BARBI PROJECT: *BARBi - Building and construction reference data library*. <http://www.barbi.no/index.jsp> (21 März 2007),

[Beer 2006] BEER, D. G.: *Systementwurf für verteilte Applikationen und Modelle im Bauplanungsprozess*, Bauhaus-Universität Weimar, Diss., 2006

[Beetz u. a. 2005] BEETZ, J. ; VAN LEEUWEN, J. ; DE VRIES, B.: An Ontology Web Language Notation of the Industry Foundation Classes. In: SCHERER, R.J., KATRANUSCHKOV, P., SCHAPKE, S.-E. (Hrsg.): *Proceedings of 22nd CIB-W78 International Conference on Information Technology in Construction*. Dresden, 2005, S. 193–198

[Beetz u. a. 2006] BEETZ, J. ; VAN LEEUWEN, J. ; DE VRIES, B.: Gebäudeinformationsmodellierung im Kontext des Semantischen Webs. In: KOCH, C., RICHTER, T., TAUSCHER, E. (Hrsg.): *Tagungsband zum 18. Forum Bauinformatik*. Weimar, 2006, S. 175–181

[Bellifemine u. a. 2003] BELLIFEMINE, F. ; CAIRE, G. ; POGGI, A. ; RIMASSA, G.: JADE - A White Paper. In: *exp 3 Issue 3* (2003), September, 6-19. <http://jade.tilab.com/papers/2003/WhitePaperJADEEXP.pdf>

[Björk 1994] BJÖRK, B.-C.: RATAS Project - Developing an Infrastructure for Computer-Integrated Construction. In: *ASCE Journal of Computing in Civil Engineering* 8 (1994), Nr. 4, S. 401–419

[Björk 1995] BJÖRK, B.-C.: *Requirements and information structures for building product data models*, Helsinki University of Technology, Finland, Diss., 1995

[Björkhaug u. Bell 2007] BJØRKHAUG, L. ; BELL, H.: *Ifd:IFD in a Nutshell*. http://dev.ifd-library.org/index.php/Ifd:IFD_in_a_Nutshell (12. März 2008), 2007

- [Boden 1994] *Kapitel 4 What Is Creativity?* In: BODEN, M. A.: *Dimensions of Creativity*. Cambridge, Massachusetts, U.S. : MIT Press, 1994, S. 75–118
- [Boehm 1988] BOEHM, B. W.: A Spiral Model of Software Development and Enhancement. In: *IEEE Computer* 21 Issue 5 (1988), S. 61–72
- [Booch 1995] BOOCH, G.: *Objektorientierte Analyse und Design*. Bonn : Addison-Wesley, 1995
- [Borrmann 2007] BORRMANN, A.: *Computerunterstützung verteilt-kooperativer Bauplanung durch Integration interaktiver Simulationen und räumlicher Datenbanken*, Technische Universität München, Diss., 2007
- [Braunes u. Donath 2006] BRAUNES, J. ; DONATH, D.: Computergestützte Planung im Bestand. Von der Digitalen Bestandserfassung zur Planungsunterstützung im CAAD. In: *Tagungsband des 17. Internationalen Kolloquiums über Anwendungen der Informatik und der Mathematik in Architektur und Bauwesen (IKM)*. Weimar, 2006, S. CD-ROM
- [Brodbeck 1995] BRODBECK, K.-H.: *Entscheidung zur Kreativität*. Darmstadt : Wiss. Buchgesellschaft, 1995
- [Brown u. Chandrasekaran 1989] BROWN, D. C. ; CHANDRASEKARAN, B.: *Design problem solving. Knowledge structures and control strategies*. London u.a. : Pitman u.a., 1989
- [Brugali 2000] BRUGALI, K. D. ans S. D. ans Sycara: Frameworks and Pattern Languages: an Intriguing Relationship. In: *ACM Computing Surveys* 32 Issue 1 (2000), S. Article No. 2
- [Burkart 1997] *Kapitel Thesaurus*. In: BURKART, M.: *Grundlagen der praktischen Information und Dokumentation*. München : K.G. Saur Verlag, 1997, S. 160 – 181

- [Buschmann u. a. 1998] BUSCHMANN, F. ; MEUNIER, R. ; ROHNERT, H. ; SOMMERLAD, P. ; STAL, M.: *Pattern-orientierte Software-Architektur. Ein Pattern-System*. München : Addison-Wesley, 1998
- [Bush 1945] BUSH, V.: As We May Think. In: *The Atlantic Monthly* 176 (1) (1945), S. 101–108
- [Caglayan u. Harrison 1998] CAGLAYAN, A.K. ; HARRISON, C.G.: *Intelligente Software-Agenten. Grundlagen, Technik und praktische Anwendung im Unternehmen*. München : Hanser-Verlag, 1998
- [Ching 1996] CHING, F D. K.: *Die Kunst der Architekturgestaltung*. Augsburg : Augustus-Verlag, 1996
- [Colomb 2002] COLOMB, R. M.: *Information Spaces. The Architecture of Cyberspace*. London u.a. , UK : Springer-Verlag, 2002
- [Cramer 1993] CRAMER, J.: *Handbuch der Bauaufnahme: Aufmaß und Befund*. Stuttgart : Deutsche Verlags-Anstalt, 1993
- [Dahlenburg 1996] DAHLENBURG, A.: *Modellierung architektonischer Objekte in frühen Entwurfsphasen mittels architekturenspezifischer Fachsprache*, Hochschule für Architektur und Bauwesen Weimar, Diss., 1996
- [Demeyer u. a. 2000] DEMEYER, S. ; DE HONDT, K. ; STEYAERT, P.: Consistent Framework Documentation with Computed Links and Framework Contracts. In: *ACM Computing Surveys* 32 Issue 1 (2000), S. Article No. 34
- [Donath 2003] DONATH, D.: Die Auseinandersetzung mit dem Bauwerk - Notwendigkeiten im Planen und Bauen. In: *Schriften der Bauhaus-Universität Weimar* 115 (2003), S. 123–132
- [Donath u. Petzold 2005] *Kapitel D2 Bestandsorientierte Unterstützung der Planung*. In: DONATH, D. ; PETZOLD, F.: *SFB 524 Werkstoffe und Konstruktionen für die Revitalisierung von Bauwerken. Bericht der 2. Förderperiode*

- 07'02-06'05 und Antrag auf Finanzierung der 3. Förderperiode 07'05-06'09.*
Bauhaus-Universität Weimar, 2005, S. 3: 547–612
- [Dörner 1976] DÖRNER, D.: *Problemlösen als Informationsverarbeitung.* Stuttgart
: Kohlhammer, 1976
- [Drodowski 1991] DRODOWSKI, R. (Red.): *Duden. Die deutsche Rechtschreibung.*
20. Auf. Mannheim u.a. : Meyers, 1991
- [Eastman 1999] EASTMAN, C.M.: *Building Product Models. Computer Environ-
ments Supporting Design and Construction.* Boca Raton, Florida, U.S. : CRC
Press, 1999
- [Elmasri u. Navathe 2002] ELMASRI, R. ; NAVATHE, S. B.: *Grundlagen von Da-
tenbanksystemen.* München : Addison-Wesley, 2002
- [Favre 2003] FAVRE, L.: *UML and the Unified Process.* Hershey : IRM Press,
2003
- [Ferber 1995] FERBER, J.: *Multi-Agent Systems. An Introduction to Distributed
Artificial Intelligence.* Harlow, England, UK : Addison-Wesley, 1995
- [Gamma u. a. 2004] GAMMA, E. ; HELM, R. ; JOHNSON, R. ; VLISSIDES, J.: *Ent-
wurfsmuster. Elemente wiederverwendbarer objektorientierter Software.* Mün-
chen : Addison-Wesely, 2004
- [Gehre u. a. 2006] GEHRE, A. ; KATRANUSCHKOV, P. ; WIX, J. ; BEETZ, J.: *Inteli-
Grid Deliverable D31: Ontology Specifications / The InteliGrid Consortium,
c/o University of Ljubljanna. Version: 2006. <http://www.inteligrid.com>(6.
März2007). 2006. – Forschungsbericht*
- [Gero 1999] *Kapitel Constructive memory in design thinking.* In: GERO, J. S.:
Design Thinking Research Symposium: Design Representation. MIT Press,
1999, S. I.29–35

- [Gessmann 2005] GESSMANN, R.: Ein internetbasiertes, digitales Gebäudebuch als Datenrepositorium. In: SCHLEY, F., WEBER, L. (Hrsg.): *Tagungsband zum 17. Forum Bauinformatik*. Cottbus, 2005, S. 1–8
- [GISMO Projekt 2008] GISMO PROJEKT: *GISMO - Ganzheitliche Integration von Sanierung und Modernisierung*. <http://www.gismo-projekt.de/> (18. März 2008), 2008
- [Grant 2008] GRANT, R.: *IFD Library White Paper*. <http://www.ifd-library.org/index.php/Overview>, 2008
- [Grønbaek u. Wiil 1997] GRØNBÆK, K. ; WIIL, U.K.: Towards a common reference architecture for open hypermedia. In: *Journal of Digital Information* 1 Issue 2 (1997), Nr. 7, S. Article No. 7
- [Grüning u. Kühn 2004] GRÜNING, R. ; KÜHN, R.: *Entscheidungsverfahren für komplexe Probleme*. Berlin u.a. : Springer-Verlag, 2004
- [Gruber 1993] GRUBER, T.R.: A translation approach to portable ontologies. In: *Journal of Knowledge Aquisition* 5 (1993), Nr. 2, S. 199–220
- [Hannus u. a. 1995] HANNUS, M. ; KARSTILA, K. ; TARANDI, V.: Requirements on standardised building product data models. In: SCHERER, R.J. (Hrsg.): *Proceedings of the European Conference on Product and Process Modelling in the Building Industry (ECPM) '94*. Rotterdam : A.A.Balkema, 1995
- [Hartmann 2000] HARTMANN, D.: *Objektorientierte Modellierung in Planung und Konstruktion*. Weinheim : Wiley-VCH, 2000
- [Hauschild 2003] HAUSCHILD, T.: *Computer Supported Cooperative Work-Applikationen in der Bauwerksplanung auf der Basis einer integrierten Bauwerksmodellverwaltung*, Bauhaus-Universität Weimar, Diss., 2003
- [Hauschild u. a. 2003] HAUSCHILD, T. ; HÜBLER, R. ; BORRMANN, A.: Techniken der Verwaltung dynamischer digitaler Bauwerksmodelle für Revitalisierungsvorhaben. In: GÜRLEBECK, K. (Hrsg.) ; HEMPEL, L. (Hrsg.) ; KÖNKE,

- C. (Hrsg.): *Proceedings of the International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering (IKM 2003)*. Weimar, 2003, S. CD-ROM
- [Hübler 1974] HÜBLER, R.: *Zur rechnerinternen Darstellung baulicher Objekte im Rahmen des automatengestützten konstruktiven Entwicklungsprozesses*, Hochschule für Architektur und Bauwesen Weimar, Diss., 1974
- [Hübler 2005] *Kapitel D3* Digitales Bauwerksmodell als Grundlage der Prozessintegration. In: HÜBLER, R.: *SFB 524 Werkstoffe und Konstruktionen für die Revitalisierung von Bauwerken. Bericht der 2. Förderperiode 07'02-06'05 und Antrag auf Finanzierung der 3. Förderperiode 07'05-06'09*. Bauhaus-Universität Weimar, 2005, S. 3: 613–668
- [Hübler u. a. 2003] HÜBLER, R. ; HAUSCHILD, T. ; WILLENBACHER, H.: Distributed Cooperative Building Models for Revivification of Buildings. In: *Schriften der Bauhaus-Universität Weimar* 115 (2003), S. 141–148
- [HOAI 1996] HOAI: *Verordnung über die Honorare für Leistungen der Architekten und Ingenieure in der Fassung der Fünften ÄnderungsVO*. Düsseldorf, 1996
- [Holzinger 2001] HOLZINGER, A.: *Basiswissen Multimedia. Band 2: Lernen*. Würzburg : Vogel, 2001
- [Hommerich u. a. 2005] HOMMERICH, C. ; HOMMERICH, N. ; RIEDEL, F. ; NORDRHEIN-WESTFAHLEN, Architektenkammer (Hrsg.): *Zukunft der Architekten - Berufsbild und Märkte. Eine Untersuchung im Auftrag der Architektenkammer Nordrhein-Westfalen*. Bergisch Gladbach : <http://www.aknw.de/data/aktuelles/detail/1129284703-6140320.pdf> (02. Juni 2008), 2005 <http://www.aknw.de/data/aktuelles/detail/1129284703-6140320.pdf> (02. Juni 2008)

- [Horn u. Reinke 2002] HORN, E. ; REINKE, T.: *Softwarearchitektur und Softwarebauelemente. Eine Einführung für Softwarearchitekten*. München : Hanser-Verlag, 2002
- [IAI 2000] IAI, (International Alliance for Interoperability): *IFC Technical Guide*. Oktober 2000
- [IAI 2004] IAI, (International Alliance for Interoperability): *IFC Model Implementation Guide*. 19. März 2004
- [ICIS 2004] ICIS, (International Construction Information Society): *TC59 / SC13 / WG6 - Framework for Object-Oriented Information Exchange*. <http://www.icis.org/index.php?page=tc/index.php> (7.Oktober 2007), 2004
- [Joedicke 1976] JOEDICKE, J.: *Angewandte Entwurfsmethodik für Architekten*. Stuttgart : Krämer, 1976
- [Johnson 1992] JOHNSON, R. E.: Documenting Frameworks using Patterns. In: *ACM SIGPLAN Notices* 27 Issue 10 (1992), S. 63–76
- [Johnson 1997a] JOHNSON, R. E.: Components, Frameworks, Patterns. In: *ACM SOGSOFT Software Engineering Notes* 22 Issue 3 (1997), S. 10–17
- [Johnson 1997b] JOHNSON, R. E.: Frameworks = (Components+Patterns). In: *Communications of the ACM Computing Surveys* 40 Issue 10 (1997), S. 39–42
- [Junge 2008] JUNGE, R.: *Interoperabilität mit IFC - Entwicklung, Grundlagen, Schema*. Vortrag zum BIM Workshop, Bauhaus-Universität Weimar, 6. November 2008
- [Kadatz 1994] KADATZ, H.-J.: *Seemanns Lexikon der Architektur*. Seemann, 1994
- [Karbach u. Linster 1990] KARBACH, W. ; LINSTER, M.: *Wissensaquisition für Expertensysteme*. München : Hanser-Verlag, 1990

- [Katranuschkov u. a. 2003] KATRANUSCHKOV, P. ; GEHRE, A. ; SCHERER, R.J.: An ontology framework to access IFC model data:. In: *Journal of Information Technology in Construction* 8 (2003), S. 413–437
- [Knorz 1997] *Kapitel* Indexieren, Klassieren, Extrahieren. In: KNORZ, G.: *Grundlagen der praktischen Information und Dokumentation*. München : K. G. Saur Verlag, 1997, S. 120 – 140
- [Koeopf 1999] KOEPF, H.: *Bildwörterbuch der Architektur*. Stuttgart : A. Kröner Verlag, 1999
- [Krech u. a. 1992] KRECH, D. ; CRUTCHFIELD, R. S. ; LIVSON, N. ; WILSON, W. A. j. ; PARDUCCI, A. ; BENESCH, H. (Hrsg.): *Grundlagen der Psychologie*. Weinheim : Psychologie Verlags Union, 1992. – Original Copyright 1958 Krech/Crutchfield
- [Kruchten 1999] KRUCHTEN, P.: *The Rational Unified Process. An Introduction*. Addison-Wesley, 1999
- [Kubicek 1997] KUBICEK, H.: *www.Stadtinfo.de - Ein Leitfaden für die Entwicklung von Stadtinformationen im Internet*. Heidelberg : Huetig, 1997
- [Kuhlen 1997] *Kapitel* Hypertext. In: KUHLEN, R.: *Grundlagen der praktischen Information und Dokumentation*. K. G. Saur Verlag, 1997, S. 355–369
- [Larman 2005] LARMAN, C.: *Appying UML and Patterns. An Itroudction to Object-Oriented Analysis and Design and Iterative Developement*. Upper Saddle River, NJ, USA : Pearson Education, 2005
- [Laux 1997] *Kapitel* Bibliografische (Literatur-) Informationssysteme. In: LAUX, W.: *Grundlagen der praktischen Information und Dokumentation*. München : K. G. Saur Verlag, 1997, S. 453–459
- [Leser u. Naumann 2007] LESER, U. ; NAUMANN, F.: *Informationsintegration. Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen*. Heidelberg : dpunkt.verlag, 2007

- [Lömker 2006] LÖMKER, T. M.: *Plausibilität im Planungsprozess. Umbau und Umutzung als Optimierungsaufgabe.*, Bauhaus-Universität Weimar, Diss., 2006
- [Lorenz 2004] LORENZ, P.: *Entwerfen: 25 Architekten - 25 Standpunkte.* München : DVA, 2004
- [Lynch 1965] LYNCH, K.: *Das Bild der Stadt.* 2. Aufl. 2001. Basel : Birkhäuser, 1965
- [Maier 1933] MAIER, N. R. F.: An Aspect of Human Reasoning. In: *Brit. J. Psychol.* 24 (1933), S. 144–155
- [Maisberger Whiteoaks/Nemetschek 2005] MAISBERGER WHITEOAKS/NEMETSCHKEK: *Europaweite Studie: Neue Geschäftspotentiale für Architekten und Ingenieure.* Maisberger Whiteoaks, München, 2005
- [Manecke 1997] *Kapitel Klassifikation.* In: MANECKE, H.-J.: *Grundlagen der praktischen Information und Dokumentation.* München : K. G. Saur Verlag, 1997, S. 141 – 161
- [Marchionini 1997] MARCHIONINI, G.: *Information Seeking in Electronic Environments.* Cambridge, U.K. : Cambridge University Press, 1997
- [Meixner 2004] MEIXNER, U.: *Einführung in die Ontologie.* Darmstadt : Wiss. Buchgesellschaft, 2004
- [Millard u. a. 2000] MILLARD, D.E. ; MOREAU, L. ; DAVIS, H.C. ; REICH, S.: FOHM: A Fundamental Open Hypertext Model for Investigation Interoperability between Hypertext Domains. In: *Proceedings of the 11th ACM on Hypertext and Hypermedia.* San Antonio, Texas, U.S., 2000
- [Nürnberg u. Leggett 1997] NÜRNBERG, P.J. ; LEGGETT, J.J.: A Vision for Open Hypermedia Systems. In: *Journal of Digital Information* 1 Issue 2 (1997), S. Article No. 8

- [Petzold 2001] PETZOLD, F.: *Computergestützte Bauaufnahme als Grundlage für die Planung im Bestand. Untersuchungen zur digitalen Erfassung und Modellbildung.*, Bauhaus-Universität Weimar, Diss., 2001
- [Petzold u. Bürgy 2005] PETZOLD, F. ; BÜRIGY, C.: Hands free - A Wearable Surveying System for Building Surveying. In: *Proceedings of the CAAD futures Conference.* Wien, 2005
- [Petzold u. a. 2008] PETZOLD, F. ; TONN, C. ; BIMBER, O. ; GRUNDHÖFER, A. ; DONATH, D.: Spatial Augmented Reality for Architecture - Designing and planning with and within existing buildings. In: *International Journal of Architectural Computing* 6 Issue 1 (2008), S. 41–58
- [Petzold u. a. 2003] PETZOLD, F. ; WENDER, K. ; DONATH, D. ; WEFERLING, U.: Das Bauwerk als Informationscontainer in den frühen Phasen der Bauaufnahme - Ausgangspunkt für die Projektentwicklung und Entwurfsformulierung. In: GÜRLEBECK, K. (Hrsg.) ; HEMPEL, L. (Hrsg.) ; KÖNKE, C. (Hrsg.): *Proceedings of the International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering (IKM 2003).* Weimar, 2003, S. CD-ROM
- [Pfeiffer u. a. 2001] PFEIFFER, M. ; KUMMER, H. ; HAUSCHKE, P.-P. ; ZAPKE, W. ; BETHE, A. ; NORDMANN, D. ; BLACHNIK, R.: *Entwicklung von Instrumenten für den nachhaltigen Sanierungsprozess von Büro- und Produktionsgebäuden sowie öffentlichen Liegenschaften.* Abschlussbericht, FH Hannover, Juni 2001
- [Polly 1996] POLLY, A.: *Methodische Entwicklung und Integration von Produktmodellen. Bd. 4.* Shaker Verlag, 1996
- [Pretz u. a. 2003] *Kapitel 1* Recognizing, Defining, and Representing Problems. In: PRETZ, J. E. ; NAPLES, A. J. ; STERNBERG, R. J.: *The Psychology of Problem Solving.* Cambridge : Cambridge University Press, 2003, S. 3 ff
- [Riedel u. Wender 2007] RIEDEL, T. ; WENDER, K.: Realisierung von Anfragefunktionalität für einen dynamischen Bauwerksmodellverbund. In: MERKEL,

- A., SCHÜTZ, R., WIESSFLECKER, T. (Hrsg.): *Tagungsband zum 19. Forum Bauinformatik*. Graz, Österreich, 2007, S. 303–310
- [Rowe 1987] ROWE, P.G.: *Design Thinking*. 5. Auflage 1994. Cambridge, Massachusetts, USA : MIT Press, 1987
- [Royce 1970] ROYCE, W. W.: Managing the Development of Large Software Systems. In: *Proceedings IEEE WESCON*, 1970
- [Rüppel u. a. 2007] *Kapitel Agentenbasierter Modellverbund für die kooperative Gebäudeplanung*. In: RÜPPEL, U. ; LANGE, M. ; THEISS, M.: *Vernetzt-kooperative Planungsprozesse im Konstruktiven Ingenieurbau. Grundlagen, Methoden, Anwendungen und Perspektiven zur vernetzten Ingenieurkooperation*. Springer-Verlag, 2007, S. 335–356
- [Rüppel u. a. 2006] RÜPPEL, U. ; MEISSNER, U.F. ; LANGE, M.: An Agent-based Cooperation Platform For Fire Protection Planning. In: RIVARD, H., MIRESCO, E., MELHEM, H. (Hrsg.): *Proceedings of the 10th International Conference on Computing in Civil & Building Engineering (ICCCBE)*. Montreal, Kanada, 2006, S. CD-ROM
- [Rüppel 2007] RÜPPEL, U. (Hrsg.): *Vernetzt-kooperative Planungsprozesse im Konstruktiven Ingenieurbau. Grundlagen, Methoden, Anwendungen und Perspektiven zur vernetzten Ingenieurkooperation*. Heidelberg : Springer-Verlag, 2007
- [Rumbaugh u. a. 1993] RUMBAUGH, J. ; BLAHA, M. ; PREMERLANI, W. ; EDDY, F. ; LORENSEN, W.: *Objektorientiertes Modellieren und Entwerfen*. München : Hanser-Verlag, 1993
- [Schmidt u. Buschmann 2003] SCHMIDT, D. C. ; BUSCHMANN, F.: Patterns, Frameworks, and Middleware: Their Synergistic Relationships. In: *Proceedings of the 25th International Conference on Software Engineering*. Portland, Oregon, 2003, S. 694–704

- [Schmidt 1989] SCHMIDT, W.: *Das Raumbuch als Instrument denkmalpflegerischer Bestandsaufnahme und Sanierungsplanung*. München : Bayrisches Landesamt für Denkmalpflege, 1989
- [Schmitz u. Krings 2004] SCHMITZ, H. ; KRINGS, E.: *Baukosten 2004, Bd.1: Instandsetzung, Sanierung, Modernisierung, Umnutzung*. Essen : Verlag für Wirtschaft und Verwaltung Wingen, 2004
- [Schwarte 2002] SCHWARTE, J.: *Das Raumbuch als Werkzeug zur Informations- und Kostensteuerung*, TU Braunschweig, Diss., 2002
- [SFB 524 2005] SFB 524, (Sonderforschungsbereich 524): *Werkstoffe und Konstruktionen für die Revitalisierung von Bauwerken. Bericht der 2. Förderperiode 07'02-06'05 und Antrag auf Finanzierung der 3. Förderperiode 07'05-06'09*. Bauhaus-Universität Weimar, 2005
- [SFB 524 2006] SFB 524, (Sonderforschungsbereich 524): *Werkstoffe und Konstruktionen für die Revitalisierung von Bauwerken. Kurzbericht zur Auslaufperiode 07'05-06'06*. Bauhaus-Universität Weimar, 2006
- [Stachowiak 1973] STACHOWIAK, H.: *Allgemeine Modelltheorie*. Wien : Springer-Verlag, 1973
- [Starke 2002] STARKE, G.: *Effektive Software-Architekturen. Ein praktischer Leitfaden*. München : Hanser-Verlag, 2002
- [Stary 1996] STARY, C.: *Interaktive Systeme. Software-Entwicklung und Software-Ergonomie*. Braunschweig : Vieweg, 1996
- [Steinmann 1997] STEINMANN, F.: *Modellbildung und computergestütztes Modellieren in frühen Phasen des architektonischen Entwurfs*, Bauhaus-Universität Weimar, Diss., 1997
- [Störrle 2005] STÖRRLE, Harald: *UML 2 für Studenten*. München : Pearson Studium, 2005

- [Taube 1998] TAUBE, W.: *Qualitativ hochwertige Stadtinformationssysteme. Zur Strukturierung des Informationsraumes*. Aachen : Shaker Verlag, 1998
- [Thurow 2004] THUROW, T.: *Untersuchungen zur computergestützten Geometrieabbildung und -verwaltung in der Bauaufnahme*, Bauhaus-Universität Weimar, Diss., 2004
- [Tonn 2007] TONN, C.: Augmentierte Bestandsplanung - Farb- und Materialgestaltung direkt vor Ort. In: MERKEL, A., SCHÜTZ, R., WIESSFLECKER, T. (Hrsg.): *Tagungsband zum 19. Forum Bauinformatik*. Graz, Österreich, 2007, S. 179–186
- [Tonn u. Donath 2006] TONN, C. ; DONATH, D.: Color, Material and Light in the Design Process - A Software Concept. In: *Proceedings of the 10th International Conference on Computing in Civil & Building Engineering (ICCCBE)*. Montreal, Kanada, 2006, S. CD-ROM
- [Urchs 2002] URCHS, Max: *Maschine, Körper, Geist. Eine Einführung in die Kognitionswissenschaft*. Frankfurt / Main : Vittorio Klostermann, 2002
- [van Steenis 1992] VAN STEENIS, H.: *Informationssysteme - wie man sie plant, entwickelt und nutzt. Ein Leitfaden für effiziente, und benutzerfreundliche Informationssysteme*. München u.a. : Hanser-Verlag, 1992
- [Vester 1999] VESTER, F.: *Die Kunst vernetzt zu denken. Ideen und Werkzeuge für einen neuen Umgang mit Komplexität*. Stuttgart : Deutsche Verlags-Anstalt, 1999
- [von Both 2004] VON BOTH, P.: *Ein systemisches Projektmodell für eine kooperative Planung komplexer Unikate*, Universität Karlsruhe, Diss., 2004
- [von Both u. Kohler 2004] VON BOTH, P. ; KOHLER, N.: Eine virtuelle Lebenszyklus-Plattform für den Baubereich. In: *Thesis - wissenschaftliche Zeitschrift der Bauhaus-Universität Weimar* 50 (2004), Nr. 1, S. 42–49

- [Wangerin 1992] WANGERIN, G.: *Bauaufnahme: Grundlagen, Methoden, Darstellung*. Braunschweig u.a. : Vieweg, 1992
- [Watson 2008] WATSON, D.: *IFD Library Use Cases*. IFD library Group Website (<http://www.ifd-library.com>) (25. April 2008), 2008
- [Weal u. a. 2001] WEAL, M.J. ; HUGHES, G.V. ; MILLARD, D.E. ; MOREAU, L.: Openhypermedia as a navigational interface to ontological information spaces. In: *Proceedings of the 12th ACM Conference on Hypertext and Hypermedia*. Århus, Dänemark, 2001, S. 227–236
- [Wiechmann 1981] WIECHMANN, H. H.: *Modernisierungshandbuch für Architekten*. Bonn - Bad Godesberg : Bundesministerium für Raumordnung, Bauwesen u. Städtebau, 1981
- [Wiil 1997] WIIL, U.K.: Open Hypermedia: Systems, Interoperability and Standards. In: *Journal of Digital Information* 1 Issue 2 (1997), S. Editorial
- [Willenbacher 2002] WILLENBACHER, H.: *Interaktive verknüpfungsbasierte Bauwerksmodellierung als Integrationsplattform für den Bauwerkslebenszyklus*, Bauhaus-Universität Weimar, Diss., 2002
- [Willenbacher u. Hübler 2004] WILLENBACHER, H. ; HÜBLER, R.: Intelligent Link Management for the Support of Integration in Building Life Cycle. In: *Proceedings of the 9th International Conference on Computing in Civil & Building Engineering (ICCCBE)*. Weimar, 2004
- [Wooldridge u. Jennings 1995] WOOLDRIDGE, M. ; JENNINGS, N.: Intelligent Agents: Theory and Practice. In: *Knowledge Engineering Review* 10 Issue 2 (1995), S. 115–152
- [Ziegler u. Beinhauer 2007] ZIEGLER, J. ; BEINHAUER, W.: *Interaktion mit komplexen Informationsräumen*. München u.a. : Oldenbourg, 2007

LITERATURVERZEICHNIS

[Zimmermann u. Campillo 2003] *Kapitel 8* Motivating Self-Regulated Problem Solvers. In: ZIMMERMANN, B. J. ; CAMPILLO, M.: *The Psychology of Problem Solving*. Cambridge : Cambridge University Press, 2003, S. 233 ff

ANHANG A.

ÜBERSICHT VERWENDETER MUSTER

Die Kurzdarstellung der in der Arbeit diskutierten Muster folgt der für Musterkataloge typischen „Schablone“ bzw. Gliederung in einzelne thematische Abschnitte, siehe etwa [Gamma u. a., 2004] sowie [Buschmann u. a., 1998]. Hinsichtlich der Abschnitte einer solchen Schablone existiert jedoch kein Konsens unter den Autoren von Musterkatalogen. An dieser Stelle sollen nur die elementaren Aspekte eines Musters, soweit sie zur Identifikation und zum Verständnis des Musters erforderlich sind, genannt werden. Interessierte können weitere Details jeweils den Musterkatalogen entnehmen, auf die an entsprechender Stelle verwiesen wird. Aus diesem Grund beschränkt sich die Musterbeschreibung hier auf die in der Tabelle A.1 genannten Abschnitte.

Mustername	Bezeichnung des Musters, ggf. alternativ verwendete Namen
Quellen	Verweise auf Musterkataloge, die das Muster ausführlich beschreiben
Kontext und Problem	Beschreibung des Kontextes, in dem das Muster angewendet werden kann; Beschreibung des adressierten Problems
Lösung	Beschreibung der wesentlichen Merkmale der Lösung, in der Regel durch die beteiligten Komponenten sowie ihre strukturellen Beziehungen und Interaktionen
Konsequenzen	Vor- und Nachteile der Verwendung des Musters

Tabelle A.1.: Schablone zur Musterbeschreibung

Model-View-Controller (MVC)

Mustername	<i>Model-View-Controller (MVC)</i>
Quellen	[Buschmann u. a., 1998, S. 124ff.]

Kontext und Problem

Strukturierung interaktiver Anwendungen

Bedienoberflächen langlebiger Systeme stellen eine sich ständig verändernde Komponente dar. Verschiedene Anwender des Systems stellen oft unterschiedliche Anforderungen an die Bedienkomponente. Folgende Kräfte wirken:

- Dieselbe Information wird in verschiedenen Sichten in unterschiedlicher Form dargestellt
- Datenänderungen sollen in allen Sichten sofort dargestellt werden.
- Komponenten der Bedienoberfläche sollen einfach austauschbar sein.

Lösung

Interaktive Anwendungen können in drei Arten von Komponenten für Verarbeitung, Eingabe und Ausgabe gegliedert werden.

- Das Modell als Verarbeitungskomponente enthält die Daten und die Kernfunktionalität für deren Verarbeitung. Es ist unabhängig von speziellen Realisierungen der Ein- und Ausgabe.
- Ausgabekomponenten oder Ansichten (englisch *views*) greifen auf die Daten zu und stellen sie nach einem bestimmten Prinzip dar.
- Eingabekomponenten (englisch *controllers*) empfangen Benutzereingaben und übersetzen diese in Dienstleistungsanforderungen an das Modell oder an eine Ansicht. Jeder Ansicht ist üblicherweise eine Eingabekomponente zugeordnet.

Struktur (siehe Abbildung A.1):

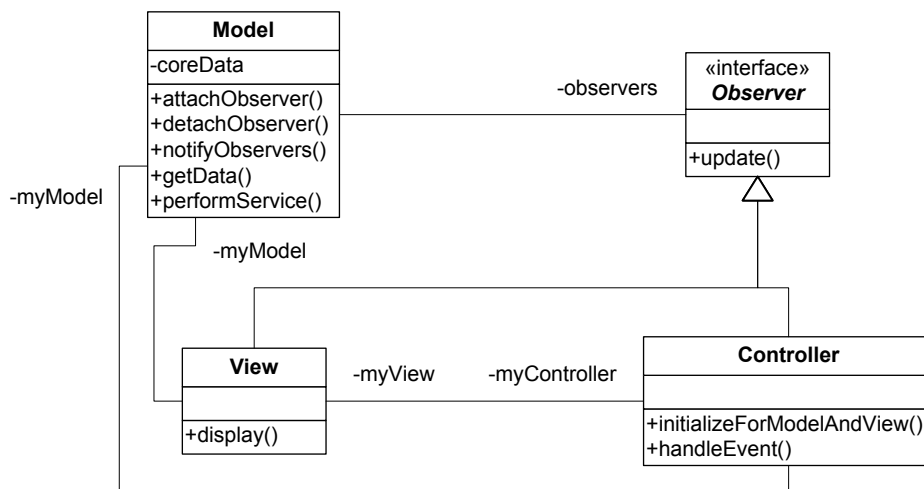


Abbildung A.1.: Komponenten des *Model-View-Controller*-Musters (UML-Klassendiagramm)

Interaktion (siehe Abbildung A.2):

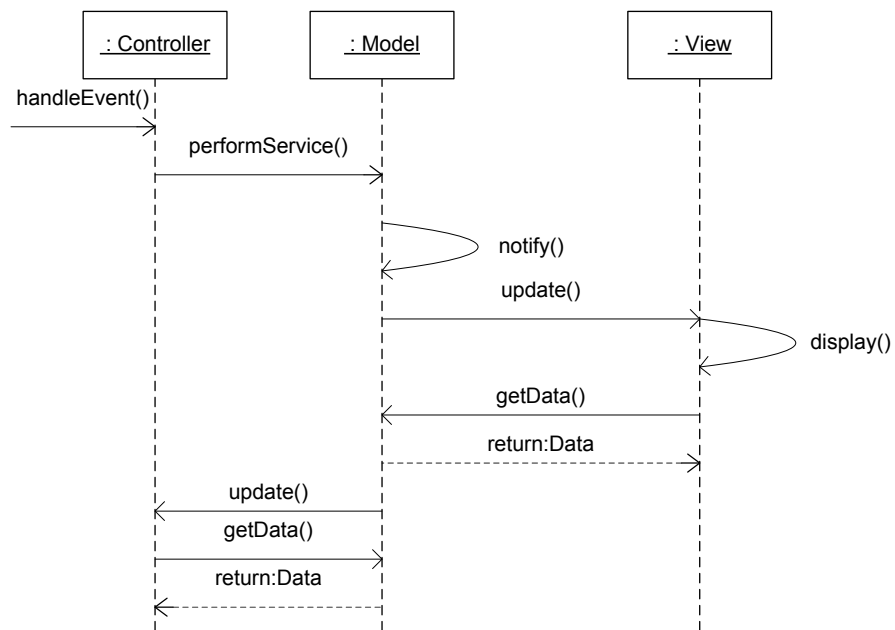


Abbildung A.2.: Interaktion der MVC-Komponenten (UML-Sequenzdiagramm)

Konsequenzen

positiv:

- Mehrere Ansichten und Eingabekomponenten können für dasselbe Modell implementiert und verwendet werden. Ansichten und Eingabekomponenten sind austauschbar.
- Die Synchronisation angemeldeter Ansichten wird durch den Benachrichtigungsmechanismus des Modells gewährleistet.

negativ:

- Die strikte Trennung von Ansichten und Eingabekomponenten bringt zusätzliche Komplexität, bei einfachen Komponenten wie Menüs jedoch keinen zusätzlichen Gewinn.
- Ansichten und Eingabekomponenten sind eng mit jeweils einem Modell gekoppelt und rufen direkt dessen Methoden auf. Änderungen am Modell bzw. dessen Schnittstelle erfordern Anpassungen aller zugehörigen Ansichten und Eingabekomponenten.
- Das Muster sieht die Verwendung eines einzigen zentralen Modells innerhalb der Anwendung vor.

Presentation-Abstraction-Control (PAC)

Mustername	<i>Presentation-Abstraction-Control (PAC)</i>
Quellen	[Buschmann u. a., 1998, S. 145 ff.]

Kontext und Problem

Strukturierung interaktiver Anwendungen

Bedienoberflächen langlebiger Systeme stellen eine sich ständig verändernde Komponente dar. Innerhalb des Systems sind unterschiedliche Funktio-

nalitäten für verschiedene in sich geschlossene Aufgaben bereitzustellen. Anwender des Systems führen unterschiedliche Aufgaben aus und stellen unterschiedliche Anforderungen an die Bedienkomponente. Folgende Kräfte wirken:

- Für unterschiedliche Aufgaben werden unterschiedliche Informationen des Gesamtdatenbestandes benötigt.
- Änderungen des Gesamtdatenbestandes sollen in allen Sichten sofort dargestellt werden, sofern sie für die betreffende Sicht relevant sind.
- Änderungen an einzelnen aufgabenspezifischen Komponenten dürfen nicht das gesamte System beeinflussen

Lösung

Das gesamte System kann als Hierarchie kooperierender Agenten aufgefasst werden, die jeweils für unterschiedliche Aufgaben verantwortlich sind. Dabei realisiert jeder Agent selbst die Verwaltung der für seine Aufgaben benötigten Daten, die erforderlichen Präsentationen und Steuerungsmechanismen. Über eine allgemein gültige Schnittstelle ist die Kommunikation der Agenten zu sichern. Jeder Agent besteht dabei aus drei Komponenten:

- Die Abstraktionskomponente eines Agenten verwaltet die benötigten Daten.
- Die Präsentationskomponente realisiert die Darstellung von Informationen für den Anwender.
- Die Kontrollkomponente eines Agenten ist verantwortlich für die interne Koordination von Abstraktions- und Präsentationskomponente und die externe Kommunikation mit anderen Agenten.

Struktur (siehe Abbildung A.3):

Interaktion (siehe Abbildung A.4):

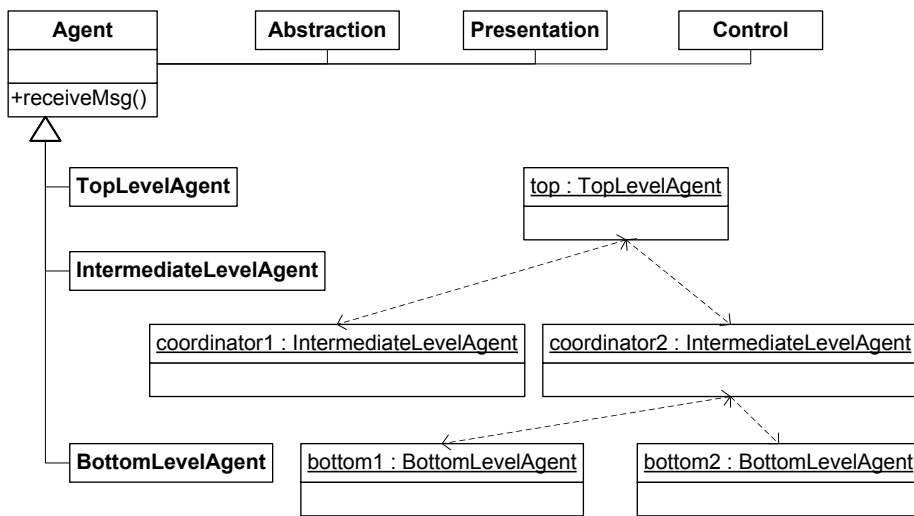


Abbildung A.3.: Komponenten des *Presentation-Abstraction-Control*-Musters (UML-Klassendiagramm)

Konsequenzen

positiv:

- Die einzelnen aufgabenspezifischen Komponenten (Agenten) sind entkoppelt und kommunizieren durch den Austausch von Nachrichten miteinander. Dies erfolgt über eine vordefinierte Schnittstelle, wobei Inhalt und Struktur der Nachrichten variabel sind.
- Das Ändern bestehender oder Hinzufügen neuer Komponenten (Agenten) hat keine unmittelbare Auswirkung auf andere beteiligte Komponenten.
- Verschiedene Belange im Anwendungsgebiet werden durch separate Agenten repräsentiert, die jeweils ihren eigenen Zustand verwalten und ihre eigene Benutzerschnittstelle zur Verfügung stellen.

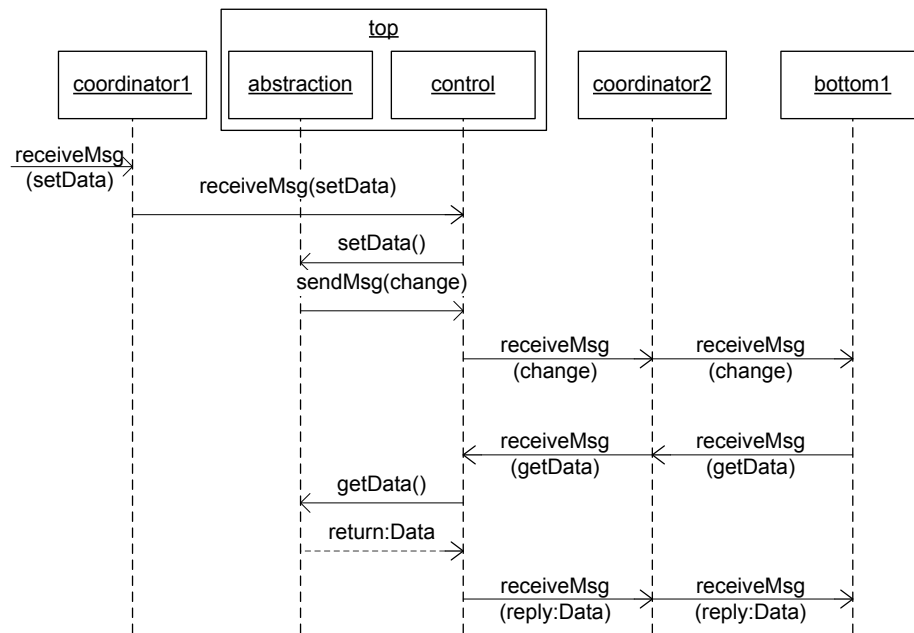


Abbildung A.4.: Interaktion der *PAC*-Komponenten (UML-Sequenzdiagramm)

negativ:

- Die Implementierung jedes einzelnen semantischen Konzeptes innerhalb eines Anwendungsbereiches als eigenständiger Agent erhöht die Systemkomplexität.
- Durch die gleichzeitige Verantwortlichkeit für die interne Koordination und die externe Kommunikation sind die Kontrollkomponenten potentiell sehr komplex.
- In Abhängigkeit von der Anzahl der Agenten erfolgt deren Kommunikation zu Lasten der Systemeffizienz.

FABRIKMETHODE, *Factory Method*

Mustername	Fabrikmethode, <i>Factory Method</i>
Quellen	[Gamma u. a., 2004, S. 131 ff.]

Kontext und Problem

Erzeugung von Objekten

Objekte sollen von einem Framework über eine vordefinierte Klassenschnittstelle erzeugt werden. Zum Zeitpunkt der Frameworkimplementierung sind die konkreten Klassen der zu erzeugenden Objekte jedoch nicht bekannt.

Folgende Kräfte wirken:

- Von abstrakten Frameworkklassen können keine Objekte erzeugt werden.
- Eine Klasse muss Objekte erzeugen, kennt jedoch die Klassen, von denen Objekte zu erzeugen sind, nicht.
- Unterklassen der erzeugenden Klasse sollen festlegen, welche Art von Objekten sie erzeugen.

Lösung

Das Problem kann mit Hilfe zweier voneinander abhängiger abstrakter Klassen gelöst werden. Eine bildet die Oberklasse für alle zu erzeugenden Objekte, die zweite die Oberklasse für alle erzeugenden Klassen. Die erzeugenden Klassen bestimmen dann jeweils, von welcher konkreten Klasse die erzeugten Objekte sind.

- Die Oberklasse der erzeugenden Klassen (**Factory**) definiert eine abstrakte Methode zur Objekterzeugung (`createProduct()`), die ein Objekt vom Typ der abstrakten Produktklasse zurückliefert. Diese Methode heißt Fabrikmethode.
- Die konkreten Unterklassen der erzeugenden Klasse überschreiben jeweils die Fabrikmethode, so dass sie ein Objekt einer konkreten Unterklasse der abstrakten Produktklasse zurück liefert.

Struktur (siehe Abbildung A.5):

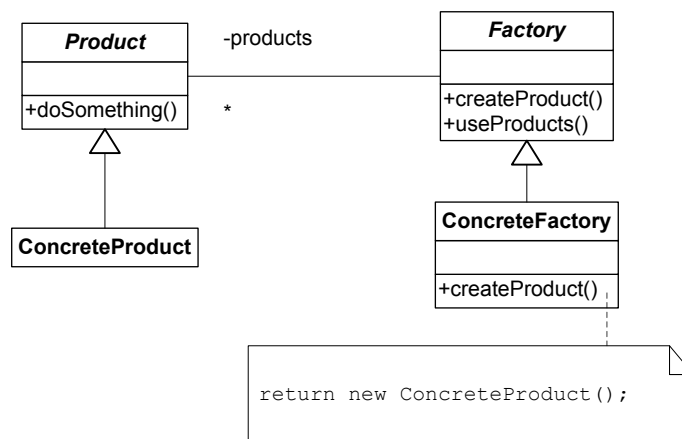


Abbildung A.5.: Komponenten des Musters *Fabrikmethode* (UML-Klassendiagramm)

Konsequenzen

positiv:

- Der Frameworkcode enthält ausschließlich die Produktschnittstelle als abstrakte Klasse, ohne dass anwendungsspezifische konkrete Klassen eingebunden werden müssen.
- Unterklassen können erweiterte Versionen des zu erzeugenden Produkts einführen. Diese Erweiterungen sind prinzipiell nicht begrenzt.

negativ:

- Die erzeugende Klasse muss innerhalb einer konkreten Anwendung möglicherweise ausschließlich wegen der Fabrikmethode abgeleitet werden. Dies bedeutet zusätzliche Klassen in Form paralleler Klassenhierarchien für erzeugende und Produktklassen.

PROTOTYP, *Prototype*

Mustername	Prototyp, <i>Prototype</i>
Quellen	[Gamma u. a., 2004, S. 144 ff.]

Kontext und Problem

Erzeugung von Objekten

Objekte sollen von einem Framework über eine vordefinierte Klassenschnittstelle erzeugt werden, wobei eine Reihe von verschiedenen Objekten zu berücksichtigen ist. Zum Zeitpunkt der Frameworkimplementierung sind die konkreten Klassen der zu erzeugenden Objekte jedoch nicht bekannt. Folgende Kräfte wirken:

- Von abstrakten Frameworkklassen können keine Objekte erzeugt werden.

- Eine Klasse muss Objekte erzeugen, kennt jedoch die Klassen, von denen Objekte zu erzeugen sind, nicht.
- Parallele Hierarchien von erzeugenden Klassen und Produktklassen wie im Fabrikmethode-Muster sollen vermieden werden.
- Die Klassen der zu erzeugenden Objekte sollen zur Laufzeit spezifiziert werden.

Lösung

Neue Objekte können zur Laufzeit durch Kopieren („Klonen“) eines Exemplars einer konkreten Produktklasse erzeugt werden.

- Eine abstrakte Oberklasse für alle zu erzeugenden Objekte (**Prototype**) definiert eine Schnittstelle, um sich selbst zu klonen.
- Clienten erhalten jeweils mindestens ein Exemplar jeder konkreten Produktklasse und rufen dessen `clone()`-Methode auf, um neue Exemplare dieser Klasse zu erzeugen.

Struktur (siehe Abbildung A.6):

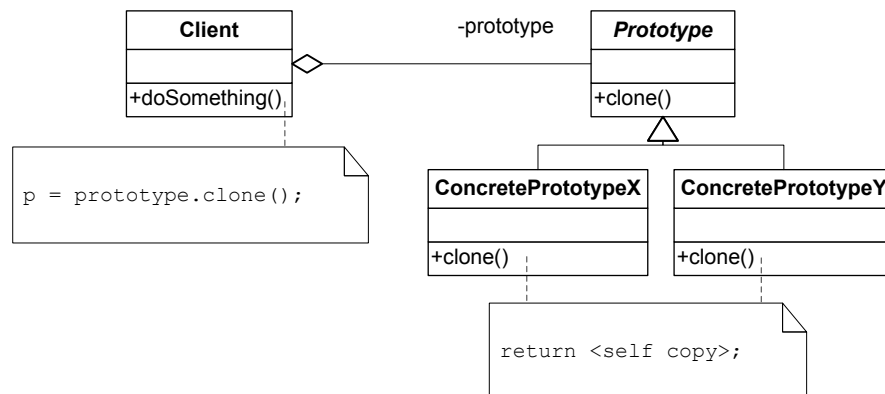


Abbildung A.6.: Komponenten des Musters *Prototyp* (UML-Klassendiagramm)

Konsequenzen

positiv:

- Neue Klassen können zur Laufzeit einfach durch das Registrieren eines Exemplars beim Clienten hinzugefügt werden.
- Parallele Klassenhierarchien von erzeugenden Klassen und Produktklassen werden vermieden, dadurch wird die Anzahl der benötigten Klassen verringert.

negativ:

- Mindestens ein Exemplar jeder verwendeten Produktklasse muss zur Laufzeit vorhanden sein, auch, wenn es im konkreten Fall nicht benötigt wird.
- Für Klassen, deren Exemplare andere Objekte enthalten, die keine Möglichkeiten zum Kopieren bieten, oder zirkuläre Referenzen enthalten, ist die Implementierung der `clone()`-Methode schwierig.

SINGLETON

Mustername	Singleton
Quellen	[Gamma u. a., 2004, S. 157 ff.]

Kontext und Problem

Erzeugung von Objekten

In manchen Situationen ist es wichtig, dass eine Klasse nur ein einziges Exemplar besitzt. Folgende Kräfte wirken:

- Es darf höchstens ein Exemplar einer bestimmten Klasse geben.
- Das Exemplar soll für alle Clienten einfach und global zugreifbar sein.
- Das Exemplar soll durch Bildung von Unterklassen erweiterbar sein.

Clients sollen diese ohne Modifikation ihres Codes verwenden können.

Lösung

Der Klasse selbst kann die Verantwortung für das Erzeugen, Verwalten und Bereitstellen ihres einzigen Exemplars übertragen werden.

- Konstruktoren der Klasse sind nur innerhalb der Klassenhierarchie sichtbar.
- Die Klasse definiert ein statisches Attribut, das eine Referenz auf das einzige Exemplar enthält.
- Clients können über eine statische Klassenmethode auf das Exemplar zugreifen.

Realisierung (siehe Listing A.1):

```
1 public class Singleton{
    private static Singleton exemplar;
3
    protected Singleton(){
5        super();
    }
7
    public static Singleton exemplar(){
9        if (exemplar == null){
            exemplar = new Singleton()
                ;
11        }
        return exemplar;
13    }
}
```

Listing A.1: Implementierung einer Singletonklasse in Java

Konsequenzen

positiv:

- Die Klasse hat strikte Kontrolle über ihr Exemplar.
- Die Klasse kann spezialisiert werden, bei Bedarf ist die Konfiguration einer Anwendung mit dem Exemplar einer erweiterten Klasse zur Laufzeit möglich.
- Derselbe Ansatz unterstützt die Bereitstellung einer festen Anzahl von Exemplaren, die größer als eins ist.

ANHANG B.

METHODEN DER KLASSE NAVISTRUCTURE ZUM ZUGRIFF AUF DIE ERSCHLIESSUNGSSTRUKTUR

Die folgende Übersicht zeigt gültige Nachrichten für Agenten vom Typ `NaviStructureManager` und korrespondierende Methoden der Klasse `NaviStructure` sowie deren Bedeutung zum Zugriff auf die Erschließungsstruktur. Dabei wird folgende Struktur verwendet:

`GÜLTIGE_NACHRICHT` - für `NaviStructureManager`

`methodenaufruf(paramerter)`; returns: `ergebnis` - für `NaviStructure`

Erläuterung der Semantik...

`GET_CONTAINERS`

`getContainers()`; returns: set of: String *containerIDs*

Liefert alle Container des Informationsraums.

`GET_CONTENT`

`getContent(InfContainer container)`; retruns: `ICContent content` Liefert den Inhalt eines Containers (als Exemplar einer eigenständigen Klasse `ICContent`).

`GET_RELATIONS`

`getRelations(set of: String containerIDs)`; returns: set of: (String *containe-*

rID, set of: Relation *relations*) *orderedResults*

Liefert alle Relationen für eine Menge von Informationscontainern, geordnet nach Containeridentifikatoren.

GET_ROOT_OF_RELATION_CONTAINS

getRootOfRelationContains(set of: String *containerIDs*); returns: set of: String *rootContainerIDs*

Liefert die Container aus einer Menge von Containern, die Wurzeln von Aggregationshierarchien sind.

GET_CLASSIFICATION_REFS

getClassificationRefs(set of: String *containerIDs*); returns: set of: (String *containerID*, set of: ClassificationRef *refs*) *orderedResults*

Liefert alle ClassificationRef-Exemplare für eine Menge von Informationscontainern, geordnet nach Containeridentifikatoren.

GET_CONTAINERS_BY_CLASSIFICATION_REF

getContainersByClassificationRef(ClassificationRef *template*); returns: set of: String *containerIDs*

Liefert alle Container für die es verknüpfte Exemplare von ClassificationRef gibt, die dem übergebenen Template entsprechen.

GET_VISUAL_OBJECT_REFS

getVisualObjectRefs(set of: String *containerIDs*); returns: set of: (String *containerID*, set of: VisualObjectRef *refs*) *orderedResults*

Liefert alle Exemplare von VisualRepresentationRef für eine Menge von Informationscontainern, geordnet nach Containeridentifikatoren.

GET_CONTAINERS_BY_VISUAL_OBJECT_REF

getContainersByVisualObjectRef(VisualObjectRef *template*); returns: set of: String *containerIDs*

Liefert alle Container für die es verknüpfte Exemplare von VisualObjectRef gibt, die dem übergebenen Template entsprechen.

GET_BM_DATA_REFS

getBMDataRefs(set of: String *containerIDs*); returns: set of:(String *containerID*, set of: BMdataRef *refs*) *orderedResults*

Liefert alle Exemplare von **BMDataRef** für eine Menge von Informationscontainern, geordnet nach Containeridentifikatoren.

GET_CONTAINERS_BY_BM_DATA_REF

getContainersByBMDataRef(BMDataRef *template*); returns: set of String *containerIDs*

Liefert alle Container für die es verknüpfte Exemplare von **BMDataRef** gibt, die dem übergebenen Template entsprechen.

GET_DOMAIN_INFO

getDomainInfo(set of: String *containerIDs*); returns: DomainInfo *info*

Liefert für eine Menge von Containern Informationen zu verfügbaren Domänenmodellen und dafür definierten Eigenschaftslinks als Exemplar einer eigenständigen Klasse **DomainInfo**.

GET_PROPERTY_LINKS

getPropertyLinks(); returns: set of: (String *domainName*, set of:PropertyLink *links*) *orderedResults*

Liefert alle für den Informationsraum definierten Eigenschaftslinks geordnet nach den Namen der Domänenmodelle.

GET_PROPERTY_LINKS_BY_BM_DATA_REF

getPropertyLinksByBMDataRef(BMDataRef *template*); returns: set of: PropertyLink *links*

Liefert alle Eigenschaftslinks, deren **source**-Referenz dem übergebenen Template entspricht.

GET_PROPERTY_LINK_VALUES

getPropertyLinkValues(PropertyLink *link*); returns: set of:(String *containerID*, set of: (BMDataRef *src*, set of: BMDataRefs *dest*)) *orderedResults*

ANHANG B. METHODEN DER KLASSE NAVISTRUCTURE ZUM ZUGRIFF AUF
DIE ERSCHLIESSUNGSSTRUKTUR

Liefert die Auswertungsergebnisse eines Eigenschaftslinks geordnet nach Informationscontainer - **source**-Referenz - **destination**-Referenzen.

GET_CONTAINERS_HAVING_PROPERTY_LINK_VALUE

getContainersHavingPropertyLinkValue(PropertyLink *link*); returns: set of:
String *containerIDs*

Liefert alle Informationscontainer, für die die Auswertung des übergebenen Eigenschaftslinks nicht null ist.

ANHANG C.

ÜBERSICHT ZU PROZESSORENTYPEN UND GÜLTIGE NACHRICHTEN

Die Auswertung der einzelnen Referenzknotentypen der Erschließungsstruktur erfolgt jeweils über korrespondierende Prozessorenklassen. Die folgende Übersicht zeigt zu jedem Referenzknotentyp die für den entsprechenden Prozessor gültigen Nachrichten und deren Bedeutung.

CLASSIFICATIONREF / CLASSIFICATIONREFPROCESSOR

GET_CONCEPT_NAME - REPLY: String *name*

Ermittelt den Namen des referenzierten Konzepts.

GET_CONCEPT_DESCRIPTION - REPLY: String *description*

Ermittelt eine Beschreibung des referenzierten Konzepts, sofern in der referenzierten Klassifikation angegeben.

GET_CONCEPT_PATH - REPLY: set of: String *pathFromRoot*

Ermittelt alle übergeordneten Konzepte ausgehend von der Wurzel der Klassifikationshierarchie bis zum tatsächlich referenzierten Konzept.

GET_CONCEPT_INFO - REPLY: ConceptInfo *info*

Übergibt die drei oben genannten Informationen als Exemplar eines Objekts der Klasse ConceptInfo.

BMDATAREF / BMDATAREFPROCESSOR

GET_OBJECT_NAME - REPLY: String *name*

Ermittelt den Namen des referenzierten Objekts.

GET_OBJECT_DESCRIPTION - REPLY: String *description*

Ermittelt eine Beschreibung des referenzierten Objekts, sofern dies vom jeweiligen Domänenmodell vorgesehen ist.

GET_OBJECT_TO_STRING - REPLY: String *objectToString*

Ermittelt eine Zeichenkette, die das referenzierte Objekt repräsentiert, analog zur `toString()`-Methode in Java, sofern dies unterstützt wird.

GET_OBJECT_INFO - REPLY: ObjectInfo *info*

Übergibt die drei oben genannten Informationen als Exemplar eines Objekts der Klasse `ObjectInfo`.

BMQUERYREF / BMQUERYREFPROCESSOR

PROCESS_QUERY

Wertet die referenzierte formale Abfrage aus.

ANHANG D.

PROTOTYPEN

D.1. ERSCHLIESSUNGSSTRUKTUR

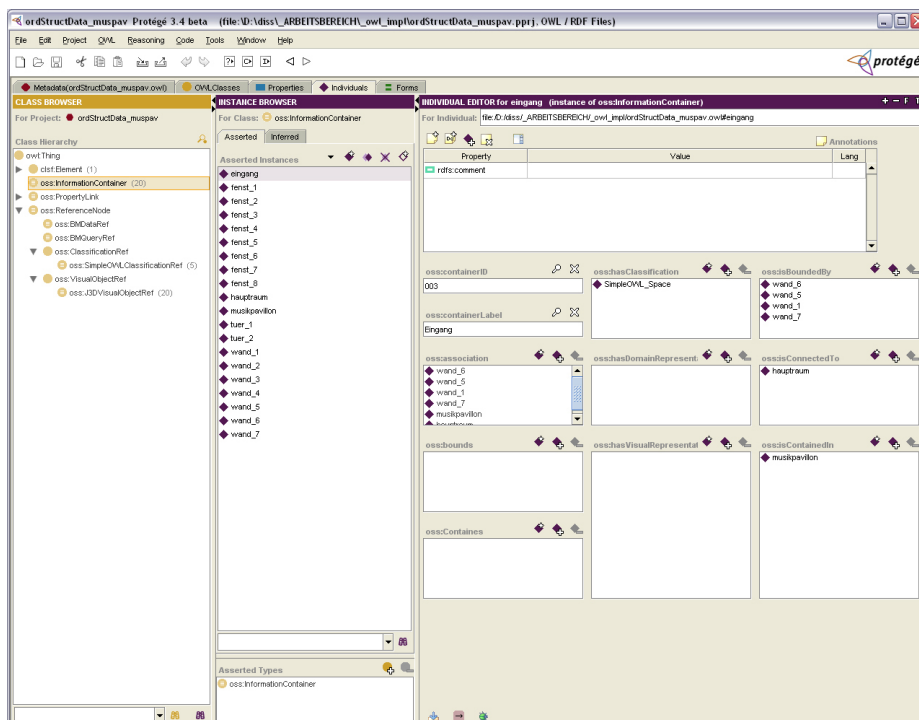


Abbildung D.1.: Bearbeitung der Erschließungsstruktur in Protégé (gelb: Schema, violett: Daten)

Realisierte Aspekte:

- Formale, maschinenlesbare Repräsentation des konzeptuellen Schemas für die Erschließungsstruktur
- Exemplarische Spezifikation eines Vokabulars für typische Raum- und Bauteiltypen von Gebäuden
- Exemplarische Ausprägung des konzeptuellen Schemas für ein einfaches Gebäude einschließlich der Referenzierung o. g. Vokabulars.

Verwendete Technologien und Werkzeuge:

- *Web Ontology Language (OWL)*¹ – formale Beschreibungssprache auf der Basis der *Extended Markup Language (XML)* zur Repräsentation konzeptueller Schemata und zugehöriger Daten
- *Protégé Ontology Editor*² – freier Editor zur Erstellung, Bearbeitung und Prüfung von OWL-Dokumenten.

Die folgenden drei Listings zeigen exemplarisch Definitionen des Datenschemas für die Klasse `InformationContainer` und die Beziehung `hasVisualRepresentation` zur Zuordnung von Referenzen auf Szeneobjekte sowie die Beziehung `bounds` zur Beschreibung topologischer Beziehungen zwischen Informationscontainern. Das vierte Listing zeigt exemplarisch ein Objekt bzw. Individuum der Klasse `InformationContainer`.

¹ siehe <http://www.w3.org/2004/OWL/>

² siehe <http://protege.stanford.edu/>

```
<owl:Class rdf:ID="InformationContainer">
2   <owl:equivalentClass>
      <owl:Class>
4         <owl:intersectionOf rdf:parseType="
          Collection">
              <owl:Restriction>
6                 <owl:onProperty rdf:resource="#
                  containerID"/>
                    <owl:cardinality rdf:datatype="&
                      xsd:int">1</owl:cardinality>
8                 </owl:Restriction>
              <owl:Restriction>
10                <owl:onProperty rdf:resource="#
                  containerLabel"/>
                    <owl:cardinality rdf:datatype="&
                      xsd:int">1</owl:cardinality>
12                </owl:Restriction>
              </owl:intersectionOf>
14          </owl:Class>
        </owl:equivalentClass>
16 </owl:Class>
```

Listing D.1: Definition der Klasse InformationContainer

```

<owl:ObjectProperty rdf:about="file:/D:ordStructSchema
    .owl#hasVisualRepresentation">
2   <rdfs:domain rdf:resource="file:/D:ordStructSchema
        .owl#InformationContainer"/>
        <rdfs:range rdf:resource="file:/D:ordStructSchema.
            owl#VisualObjectRef"/>
4 </owl:ObjectProperty>

```

Listing D.2: Definition der Beziehung hasVisualRepresentation

```

<owl:ObjectProperty rdf:about="file:/D:ordStructSchema
    .owl#bounds">
2   <owl:inverseOf rdf:resource="file:/D:
        ordStructSchema.owl#isBoundedBy"/>
        <rdfs:subPropertyOf rdf:resource="file:/D:
            ordStructSchema.owl#association"/>
4   <rdfs:comment xml:lang="de">Linkes Element (
        Bauteil = Subjekt) begrenzt rechtes Element (
            Raum = Objekt)</rdfs:comment>
        <rdfs:label xml:lang="en">bounds</rdfs:label>
6   <rdfs:label xml:lang="de">begrenzt</rdfs:label>
</owl:ObjectProperty>

```

Listing D.3: Definition der Beziehung bounds

```

1 <oss:InformationContainer rdf:ID="wand_1">
    <oss:containerID rdf:datatype="&xsd:string"
        >004</oss:containerID>
3   <oss:containerLabel rdf:datatype="&xsd:string"
        >Wand_1</oss:containerLabel>
        <oss:bounds rdf:resource="#eingang"/>
5   <oss:bounds rdf:resource="#hauptraum"/>

```

```

7     <oss:Contains rdf:resource="#tuer_2"/>
    <oss:hasVisualRepresentation rdf:resource="#
        j3do_004"/>
    <oss:hasClassification rdf:resource="#
        SimpleOWL_Wall"/>
9 </oss:InformationContainer>

```

Listing D.4: Ein Exemplar der Klasse InformationContainer

D.2. PROPERTY LINK MANAGER

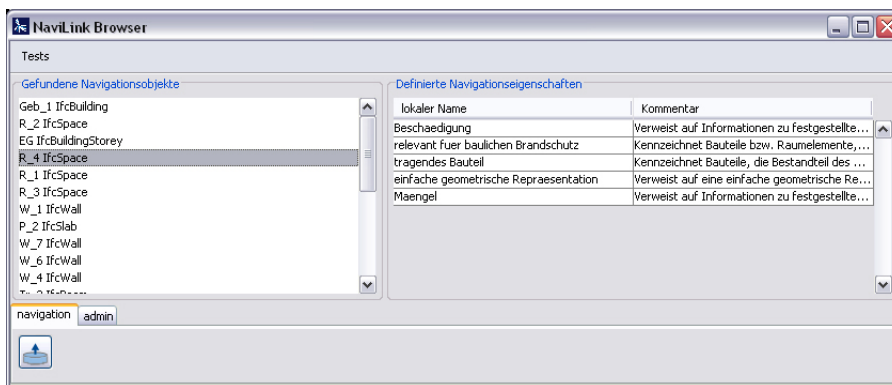


Abbildung D.2.: Benutzeroberfläche: Übersicht zu definierten Eigenschaftslinks

Realisierte Aspekte:

- Formale, maschinenlesbare Repräsentation von Anfragemustern und Eigenschaftslinks in OWL
- Einlesen von OWL-Dokumenten und Generieren eines entsprechenden Laufzeitmodells

```

C:\WINDOWS\system32\cmd.exe
<rdf:Description rdf:about="http://semantic.ddss.nl/ontologies/2006/03/ifcOWL2
x2.owl#IfcBeam">
  <owl:equivalentClass>
    <rdf:Description rdf:about="file:/D:/iww/entwicklung/eclipse_workspace/jen
aprojects/vocabulary/NavioOnt.owl#Beam">
      <owl:equivalentClass rdf:resource="http://semantic.ddss.nl/ontologies/20
06/03/ifcOWL2x2.owl#IfcBeam"/>
    </rdf:Description>
  </owl:equivalentClass>
</rdf:Description>
</rdf:RDF>
Bearbeitungszeit in s: 589.0
----- end: reloadAllDefFiles -----
----- start: buildQueryFor -----
----- start: initOntModel -----
----- end: initOntModel -----
Input-Name: IsPartOfStructuralSystem_query
Property: IsPartOfStructuralSystem_query
Bearbeitungszeit in s: 1.0
----- end: buildQueryFor -----

SELECT STRUCT(building_element: bel, structural_member: stm) FROM r IN bel.Refer
encedBy, bel IN IfcBuildingElement, stm IN IfcStructuralMember, stm_ IN r.Relate
dObjects WHERE stm=stm_;
  
```

Abbildung D.3.: Kontrollausgabe einer erzeugten Anfrage

- Persistente Verwaltung des Modells mit Hilfe einer relationalen Datenbank
- Ausprägen („Instanzieren“) von Anfragemustern auf Anforderung des Nutzers, Generieren entsprechender Anfragen zur Übergabe an ein Anfragemodul des Modellverwaltungssystems.

Verwendete Technologien und Werkzeuge:

- Programmiersprache: *Java*
- *Web Ontology Language (OWL)*
- *Jena RDF Framework*³ – Einlesen von OWL-Dokumenten in Java-Anwendungen, Generieren eines entsprechenden Laufzeitmodells
- *MySQL*⁴-Datenbank – persistente Verwaltung.

³ siehe <http://jena.sourceforge.net/>

⁴ siehe <http://www.mysql.de/>

D.3. NAVI MODEL EDITOR

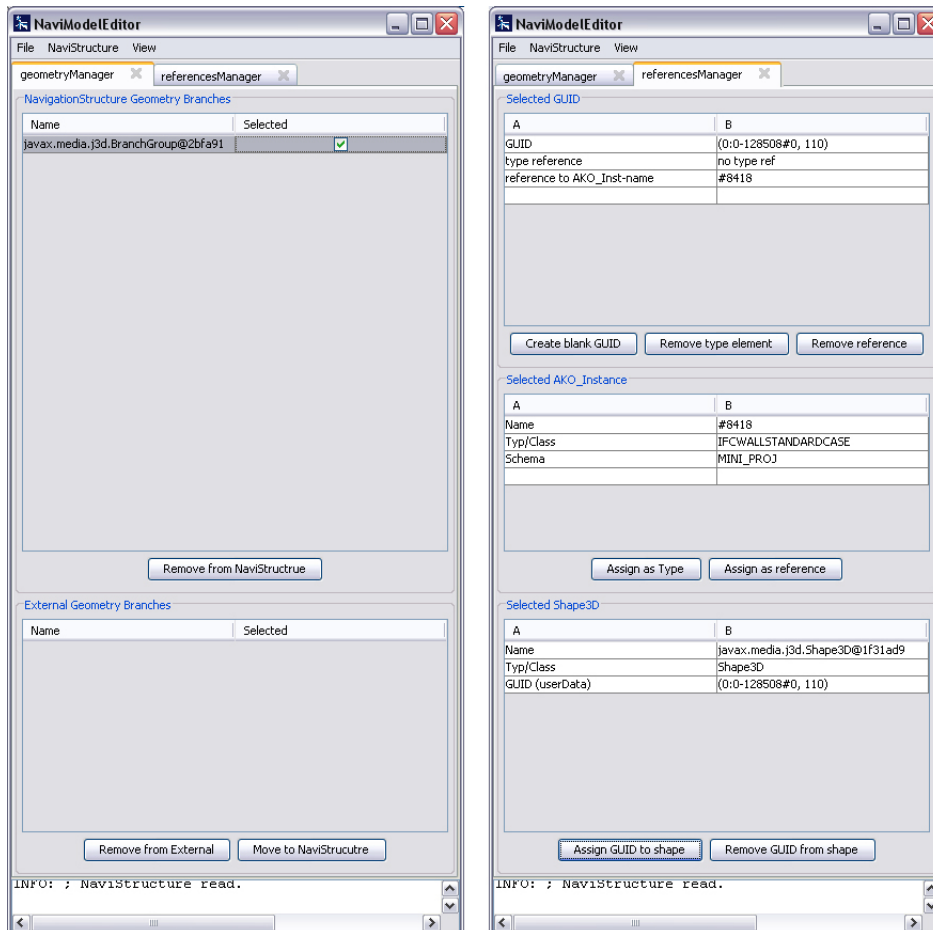


Abbildung D.4.: Übersichts- und Arbeitsfenster des Editors



Abbildung D.5.: 3D-Fenster zur Auswahl von Szeneobjekten

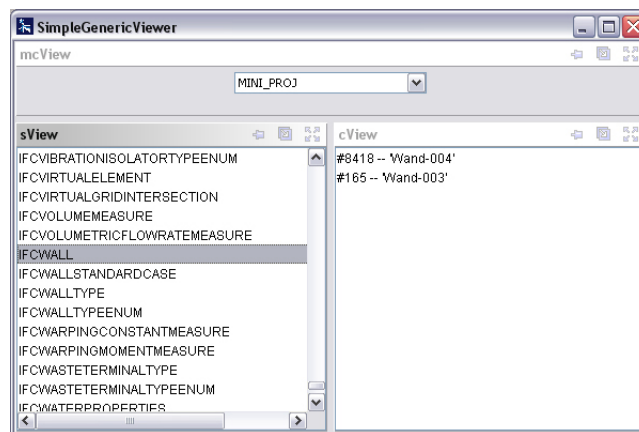


Abbildung D.6.: Fenster zur Auswahl domänenspezifischer Repräsentationen (IFC-Objekte)

Realisierte Aspekte:

- (Manuelle) Erstellung eines konzeptuellen Schemas zur Verwaltung der objektbezogenen Ordnungsstruktur einschließlich exemplarischer Daten für ein einfaches Gebäude
- Persistente Verwaltung von Schema und Daten als eigenständiges (Teil-)Modell mit Hilfe eines Dynamik unterstützenden Modellverwaltungssystems [Hauschild, 2003; Willenbacher, 2002]
- Einlesen von Geometrieinformationen aus CAD-Daten und Transformation in eine 3D-Szenenbeschreibung
- Zuordnung von Objekten einer extern verwalteten 3D-Szenenbeschreibung zu Elementen der Ordnungsstruktur
- Einlesen IFC-basierter Daten in ein exemplarisches fachspezifisches Modell
- Manuelles Zuordnen zwischen Elementen der Ordnungsstruktur, Szeneobjekten und domänenspezifischen Repräsentationen.

Verwendete Technologien und Werkzeuge:

- Programmiersprache: *Java*
- *Prototypisches Modellverwaltungssystem DYNAMO* (Server und Browser/Editor)
- *ArchiCAD*⁵ – Architekturbezogene Modellierung eines einfachen Gebäudes, Export von Geometrieinformationen und IFC-Daten
- *Java3D Framework*⁶ – Erzeugung, Verwaltung, Manipulation und Präsentation von 3D-Szenen.

⁵ siehe <http://www.graphisoft.de/>

⁶ siehe <http://java.sun.com/javase/technologies/desktop/java3d/>

D.4. ISEE – INFORMATION SEEKING ENVIRONMENT, FRAMEWORK

Realisierte Aspekte:

- Aufbau der Agentenhierarchie als Grundstruktur der Benutzerschnittstelle
- Asynchrone nachrichtenbasierte Kommunikation der Komponenten, Festlegung wesentlicher Kommunikationsabläufe zum Aufbau der Hierarchie
- Zentrale Agenten: `SessionManager`, `DefaultSubsetManager` und `-Factory`
- Vorgabe einer abstrakten Klasse als Template für Fabrik- und Viewer-Agenten.

Verwendete Technologien und Werkzeuge:

- Programmiersprache: *Java*
- *Java Agent Development Framework (JADE)*⁷.

⁷ siehe <http://jade.tilab.com/>

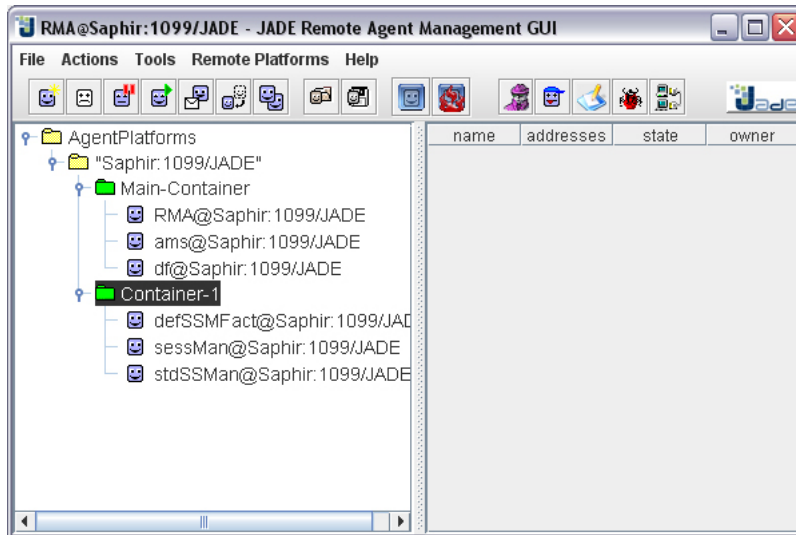


Abbildung D.7.: Grafische Oberfläche des Jade Remote Manager Agenten

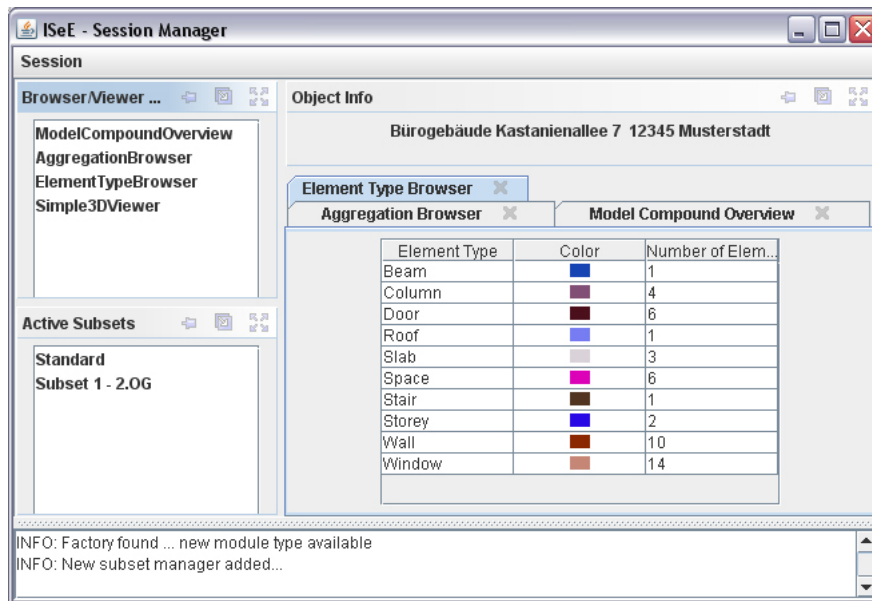


Abbildung D.8.: Grafische Oberfläche des SessionManager-Agenten

ANHANG E.

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Bei der Auswahl und Anfertigung von Material haben mir andere Personen weder entgeltlich noch unentgeltlich geholfen.

Weitere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder anderer Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Ich versichere ehrenwörtlich, dass ich nach bestem Wissen die reine Wahrheit gesagt und nichts verschwiegen habe.

Weimar, den 26. November 2008

Katrin Wender

ANHANG F.

ÜBER DIE AUTORIN

LEBENS LAUF

Persönliche Daten

Name: Katrin Wender
geboren: 17. Juli 1977 in Mühlhausen
Familienstand: ledig

Schulbildung

09/1984–08/1991 Polytechnische Oberschule, Mühlhausen
09/1991–06/1996 Tilesius Gymnasium, Mühlhausen
Abschluss: Allgemeine Hochschulreife

Studium

10/1996–12/2002 Architektur an der Bauhaus-Universität Weimar,
Vertiefungsrichtung: Computergetützte Entwurfs-
und Planungsmethoden
Abschluss: Diplom-Ingenieurin (Dipl.-Ing.)

Studienbegleitende Tätigkeit

03/2000–12/2002 Studentische Hilfskraft an den Professuren
Informatik in der Architektur (Prof. Donath) und

Informations- und Wissensverarbeitung (Prof. Hübler)
der Bauhaus-Universität Weimar

Beruflicher Werdegang

01/2003–04/2003 Wissenschaftliche Hilfskraft an den Professuren
Informatik in der Architektur und
Informations- und Wissensverarbeitung

seit 05/2003 Wissenschaftliche Mitarbeiterin an der Professur
Informations- und Wissensverarbeitung,
zwischenzeitlich an der Professur
Informatik in der Architektur

Bearbeitung der Teilprojekte D2 „Bestandsorientierte Unterstützung der Bauplanung“ und D3 „Digitales Bauwerksmodell als Grundlage der Prozessintegration“ des DFG Sonderforschungsbereiches 524 „Werkstoffe und Konstruktionen für die Revitalisierung von Bauwerken“ sowie des DFG-Einzelprojektes „Modulare, dynamische Recherchewerkzeuge zur Unterstützung der kooperativ-integrativen Entscheidungsfindung im Bauwerkslebenszyklus“

PUBLIKATIONEN

WENDER, K.: Strategien der Informationsrecherche und -aufbereitung von komplexen Gebäudedaten in der Bestandsplanung. In: KAAPKE, K., WULF, A. (Hrsg.): *Tagungsband zum 15. Forum Bauinformatik*. Hannover : Shaker Verlag, Aachen, 2003, S. 315–322

PETZOLD, F. ; WENDER, K. ; DONATH, D. ; WEFERLING, U.: Das Bauwerk als Informationscontainer in den frühen Phasen der Bauaufnahme - Ausgangspunkt für die Projektentwicklung und Entwurfsformulierung. In: GÜRLEBECK, K. (Hrsg.) ; HEMPEL, L. (Hrsg.) ; KÖNKE, C. (Hrsg.): *Proceedings of the International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering (IKM 2003)*. Weimar, 2003, S. CD-ROM

WENDER, K. ; DONATH, D. ; PETZOLD, F.: Preparation and Provision of Building Information for Planning within Existing Built Contexts. In: *Proceedings of the 9th International Conference on Computing in Civil & Building Engineering (ICCCBE)*. Weimar, 2004, S. CD-ROM

WENDER, K. ; WILLENBACHER, H. ; HÜBLER, R. ; DONATH, D.: A Modular Navigation Layer for Information Retrieval in the Building Life Cycle. In: *Proceedings 23rd eCAADe Conference*. Lissabon, Portugal, 2005, S. 581–588

WENDER, K. ; WILLENBACHER, H.: Eine modulare Integrierende Navigations-ebene für die Informationsrecherche innerhalb des Bauwerkslebenszyklus. In: *Tagungsband zum 17. Forum Bauinformatik*. Cottbus, 2005, S. 52–59

HÜBLER, R. ; WENDER, K. ; WILLENBACHER, H.: Verknüpfungsbasierte Integra-

tion im Bauwerkslebenszyklus. In: *Schriften der Bauhaus-Universität Weimar 117*. Weimar, 2005

WENDER, K. ; WILLENBACHER, H. ; HÜBLER, R.: Information Retrieval in Building Life Cycle – Modular Navigation Tools For A Dynamic Integration Base. In: RIVARD, H., MIRESCO, E., MELHEM, H. (Hrsg.): *Proceedings of the 10th International Conference on Computing in Civil & Building Engineering (ICCCBE)*. Montreal, Kanada, 2006, S. CD-ROM

WILLENBACHER, H. ; WENDER, K. ; HÜBLER, R.: GIS-Functionality for Building Model Integration and Analysis. In: RIVARD, H., MIRESCO, E., MELHEM, H. (Hrsg.): *Proceedings of the 10th International Conference on Computing in Civil & Building Engineering (ICCCBE)*. Montreal, Kanada, 2006, S. CD-ROM

WENDER, K. ; HÜBLER, R. ; WILLENBACHER, H.: Benutzeradäquate Navigations- und Recherchewerkzeuge für dynamische verteilte digitale Bauwerksmodelle. In: *Tagungsband des 17. Internationalen Kolloquiums über Anwendungen der Informatik und der Mathematik in Architektur und Bauwesen (IKM)*. Weimar, 2006, S. CD-ROM

WENDER, K. ; VAD, J.: Realisierung von Anfragefunktionalität basierend auf einem dynamischen Partialmodellverbund. In: KOCH, C., RICHTER, T., TAUSCHER, E. (Hrsg.): *Tagungsband zum 18. Forum Bauinformatik*. Weimar, 2006, S. 229–236

WENDER, K. ; HÜBLER, R.: Towards an information seeking environment for distributed building related data: Introduction to a system concept based on a shared ontology. In: SOIBELMAN, L.; AKINCI, B. (Hrsg.): *Proceedings of the 2007 ASCE International Workshop on Computing in Civil Engineering*. Pittsburg/Pennsylvania, USA, 2007, S. 127–134

WENDER, K. ; RIEDEL, T.: Unterstützung der kontextbezogenen Informationsrecherche in komplexen Bauwerksmodellen. In: MERKEL, A., SCHÜTZ, R., WIESSFLECKER, T. (Hrsg.): *Tagungsband zum 19. Forum Bauinformatik*. Graz, Österreich, 2007, S. 187–194

- RIEDEL, T. ; WENDER, K.: Realisierung von Anfragefunktionalität für einen dynamischen Partialmodellverbund. In: MERKEL, A., SCHÜTZ, R., WIESS-FLECKER, T. (Hrsg.): *Tagungsband zum 19. Forum Bauinformatik*. Graz, Österreich, 2007, S. 303–310
- WENDER, K. ; HÜBLER, R.: Mapping context-specific vocabulary to a large-scale product model to realize a more user-centred information seeking environment. In: LIMA, C. P., BAUER, M. (Hrsg.): *Information and Knowledge Management Helping the Practitioner in Planning and Building, Proceedings of the CIB W102 3rd International Conference*. Stuttgart, 2007, S. 302–311
- WENDER, K. ; HÜBLER, R.: Enabling more human-oriented exploration of complex building information models. In: *Journal of Computing in Civil Engineering* 23, No. 2 (2009), S. 84–90